

The Ruby & Rails Ecosystem White Paper

Initial Release Version - Fall 2009



Developed and Distributed by:



InfoEther, Inc.

12801 Worldgate Drive, Suite 500

Herndon, Virginia 20170 USA

<http://infoether.com>

Table of Contents

| | | |
|--------------------------------|-------|----|
| Executive Summary | | 1 |
| Ruby & Rails | | 5 |
| Enterprise User Stories | | 9 |
| Ruby Runtimes | | 11 |
| Ruby Frameworks | | 14 |
| Community Support | | 16 |
| Conferences | | 17 |
| Training | | 20 |
| Publishing | | 22 |
| Implementation Activity | | 24 |
| Consulting | | 25 |
| Implementation Tools | | 27 |
| Databases | | 28 |
| Application Management | | 29 |
| Deployment | | 30 |
| Hosting (Applications) | | 32 |
| Hosting (Code) | | 34 |
| Technology Tools | | 36 |
| Developer Tools | | 37 |
| Analysis Tools | | 40 |
| Recap | | 41 |

Copyrights, trademarks and logos listed or depicted are the property of their respective owners

“Ruby on Rails” and “Rails” are trademarks of David Heinemeier Hansson

Photo attribution is provided & used with permission along with purchased clip art

Note: Web links to source data (instead of footnotes) are provided where possible and embedded in the text of this PDF document

This document is for educational purposes only. No investment advice is being offered. Material herein is based on firsthand experiences, interviews, publicly available information, and opinions and observations of individuals in the Ruby and Rails community or outside commentators. This is a living document and will be updated and refreshed periodically. An attempt was made to be as comprehensive as possible though space and format limitations did constrain the amount of material. Companies that cross over many of the segments are usually listed in just the segment for which they are best known. Feedback and new information can be sent to this email address for consideration for inclusion in future revisions: ecosystem@infoether.com

Executive Summary

by **Mark Gardner**
Co-founder & COO, InfoEther, Inc.

...

INTRODUCTION

A few days after Labor Day in 2008 at a Denver hotel, an acclaimed instructor running a course in computer software realized one of his pupils was a venture capitalist. When asked why he was in a class to learn “Ruby on Rails” beside a raft of Java programmers eager to reinvent themselves, the VC answered simply—all the start-ups coming to him were using it, and he wanted to find out why it was such a big deal.

A few weeks earlier, a government employee approached a keynote speaker at a local Ruby User’s Conference outside Washington, DC, and related to him that her agency was about to launch an ambitious plan to convert 40,000 Lotus Notes and ColdFusion databases to Ruby on Rails in a major move to refresh their infrastructure by moving to Open Source and away from proprietary technologies.

A few months earlier, a physician entrepreneur besieged by institutional investors to buy a stake in his hot and successful start-up turned the questioning back on them. He quizzed them on the best technology with which to build and expand his popular online site. What were their portfolio companies using? The uniform answer—the Ruby programming language and the Ruby on Rails Web application framework built on top of it.

AN UNUSUAL START

How does a computer software language invented in Japan and only widely available less than a decade ago suddenly have so much traction?

Why are so many Web sites—the basis for online commerce—racing to be built with a framework written on top of that language? A framework barely five years old originally created by a young Danish programmer.

“Better, faster, and cheaper.” That is the short answer—the classic business mantra. The long answer is a richer and more interesting tale.

The story of how this came about follows the same path as previous major advances in technology. The characters and circumstances are different, but the plot is essentially the same. This story line has reached a critical juncture. The young star laboring in obscurity has received a big break and is now going mainstream. The chasm has been crossed, and the wider world is coming to it.



RUBY

What: A dynamic, general-purpose object-oriented programming language; sometimes referred to as a “scripting” language

Timeline: Appeared in Japan in 1995; worldwide availability with English documentation in 2000

Developer Population:
[Estimated](#) < 1 million worldwide; projected to be 4 million by 2013

License: Open Source-[Public Licenses](#) (Free to download; no licensing fees; non-proprietary & cross-platform)

Creator: Yukihiro Matsumoto (aka “Matz”) & others

Audience

This paper is designed to reach a mainstream business audience—laypeople who are looking to understand the phenomenon of the Ruby language and how to profit from it either through the use and deployment of the technology or through investments. It will be of limited use to technology experts.



RUBY ON RAILS

What: A “Web application framework” built on top of the Ruby programming language

Alternative Names: RoR, Ruby-on-Rails, or simply Rails

License: Open Source-MIT (Free to download; no licensing fees; non-proprietary & cross-platform)

Use: Creating “database-driven” Web software applications

Timeline: Appeared 2004; derived from a for-profit collaborative software service called Basecamp created by the Chicago-based company 37signals

Creator: David Heinemeier Hansson (aka “DHH”) & others (Rails Core Team)

News: Major update with next release (Rails 3.0); combines Merb framework & team

“I always knew one day Smalltalk would replace Java. I just didn’t know it would be called Ruby.”

*~ Kent Beck, Creator of
“Extreme Programming”*

NOT ALL SWEETNESS & LIGHT

That is not to say this is not occurring without controversy or disagreement on the details of the change underway. When one sifts through the shrill and sometimes misinformed commentary and the various purist arguments and gets to the more balanced discussions, there is a realization that something significant is happening and for many good reasons.

There have been attempts to create metrics to chart Ruby and Rails’ growth and velocity using search data by topic, code sample postings, and book sales, as well as more concrete measures such as downloads statistics for both Ruby and Rails. However, these numbers can represent only an approximation of the activity around them and therefore cannot be used as actual indicators of adoption.

Little evidence exists of proper market surveys. Even if they were done today, surveys may not tell the true story since the knowledge of Ruby and Rails is just beginning to seep into the consciousness of the broader “Enterprise” or corporate marketplace.

At this point in time, much of what could be called marketing data that is available and of value is anecdotal. This paper is comprised of hard data, first hand knowledge, and interviews with industry practitioners.

TRANSITION TO THE MAINSTREAM

In the data gathered, it is apparent that the “Enterprise” market has its “early movers” involved in assessing the technology of Ruby and Rails. This can be clearly seen in the type of consulting projects that are emerging, the backgrounds of the people attending conferences and training classes, as well as the growth in topical publications, regional events, and users groups.

Originally the territory of young developers, hobbyists, and hackers, the Ruby and Rails landscape has expanded. Banks, government agencies, and media companies are moving to take advantage of both the efficiencies of the technology and the freedom offered by Open Source software.

Most of the companies and organizations adopting Ruby and Rails today are still technology-oriented start-ups, usually venture funded. But this is changing, and the change has become obvious to the casual observer within the last year.

RAILS GOES MAINSTREAM

The economic downturn that started in late 2008 has encouraged classic IT enterprises to move to Open Source solutions. Companies and government agencies can dispose of recurring licensing costs and transition their programmers to more productive tools that speed development and thus cut time-to-market or time-to-internal-deployment costs.

This subtle transition is noticeable by those on the front lines—consultants, trainers, and conference planners—even when you factor out the understandable cut backs that organizations have taken that do affect travel, conference attendance, project timing, etc.

DISGRUNTLED DIGERATI

A few in the Ruby community are not happy with this transition. Several of the more vocal community bloggers have expressed their displeasure with this change of state, and some are moving on to “the next big thing,” whatever that may be. These early adopters often point out limitations and problems with Rails in hopes of scaring away mainstream developers, leaving Ruby and Rails as a more fertile creative playground.

But the limited range of problems that have accompanied both Ruby and Rails are often blown out of proportion by naysayers and proponents of competitive languages and frameworks. Many of the well-known limitations are the echoes of myth and legend. The others are being addressed by a large and growing community of developers.

Recently released versions of both Ruby and Rails address many issues, and the next full releases of each—Ruby 2.0 and Rails 3.0—should overcome the majority of the often-cited issues with speed, function, and capability.

AN ECOSYSTEM EMERGES

What many of the naysayers do not grasp is that the considerable ecosystem that has now built up around both Ruby and Rails is one of the main attractions that is really fueling the mainstream transition. It means

The point of this paper is to explore the Ruby & Rails “Ecosystem” that has arisen as well as point out the considerable opportunities that remain to expand it further—opportunities for investors and entrepreneurs as well as for new consumers of the technology.

stability and support, and that is what the institutional software developers need. That means more options.

The point of this paper is to explore this ecosystem as well as point out the considerable opportunities that remain to expand it further—opportunities for investors and entrepreneurs as well as for new consumers of the technology.

COMPARISONS TO OTHERS

There is an old Flip Wilson joke that has a train conductor trying to calm down a woman who claims the drunk in the opposite seat had called her baby ugly. The conductor apologized and offered the woman a free meal in the dining car where he would also provide a banana for her monkey.

Adherents of some software languages also often see a competing language as a monkey and not a baby. They are invested in their prejudices. Hence, it does take patience and persistence to understand the differences and find the value.

To the curious newcomer, Web searches will reveal a wide range of commentary on Ruby and Rails that can be contradictory as well as confusing. This paper will try to be more even handed in pointing out the good and the bad without digging too deeply into the religious technical debates that often accompany comparisons.

Those reading this paper hopefully will want to learn and understand what the big deal is all about without getting hopelessly tied down by the minutiae.

THE PROVERBIAL “KILLER APP”

Many observers have attempted to explain the various software programming languages and their inter-relationship. These are at best apples-to-oranges comparisons. The fact is many of the languages work together, and none by itself is a sufficient solution. It depends on circumstances and resources as to the programming tools selected to address an application.

Ruby was a relatively minor player as a software language until Rails appeared in 2004. (Ruby is in fact as old as Java, both released originally in 1995. But Ruby was Open Source and did not have a large corporation behind it as Java did with Sun Microsystems.)

Rails was a very specific tool that came out at the right time to address a major area of business interest—creating Web applications that use databases—and getting them built and functioning quickly while allowing continuous changes and updates.

Rails just so happened to also tie in with the rise and use of Agile programming methodologies, which were created to rapidly speed up the creation of complex software projects.

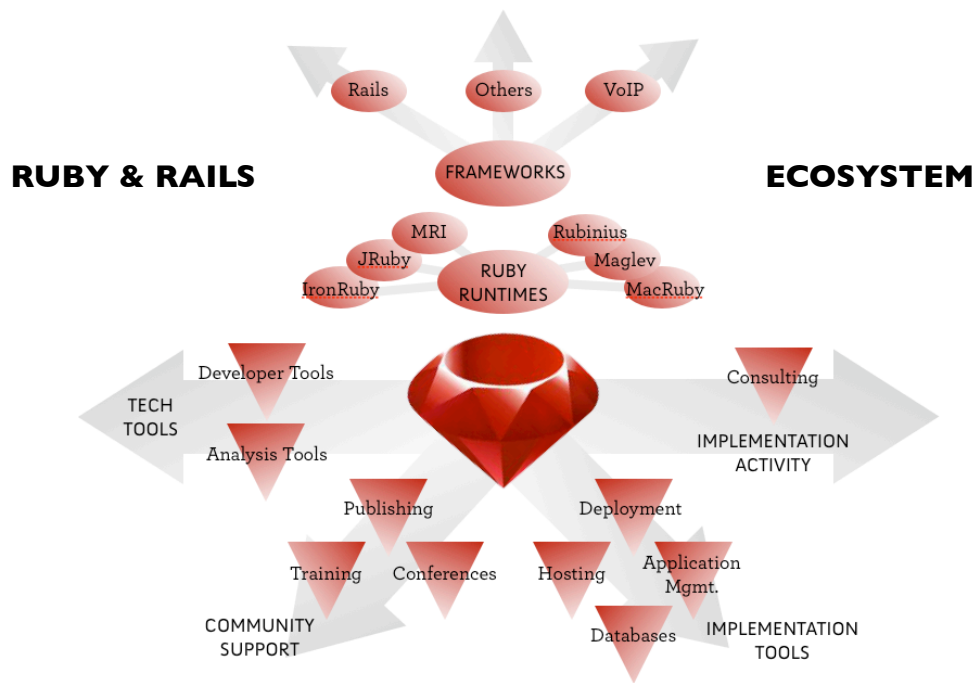
This convergence—with a little evangelizing promotion by the Rails team—made things really take off. Other languages that were similar did not have such tools of this caliber during this critical time, though attempts to (poorly) clone it developed. But Rails stole the march on everyone and achieved mind share.

Rails ascendancy brought Ruby along too, because to do more sophisticated things in conjunction with Rails, programmers needed to use Ruby to extend things. Ruby gave them some very hot capabilities, such as the ability to do “meta-programming,” which is enormously powerful.

Ruby also was programmer-friendly, meaning one did not have to fight the language—long a major problem in the software development world. This helped its popularity.

CHARTING THE ECOSYSTEM

The graphic on this page illustrates the initial organization of the Ecosystem as it is discussed in this paper. It is by no means complete and will be refreshed in future versions of this document. For now, it’s a place to start.



Ruby & Rails

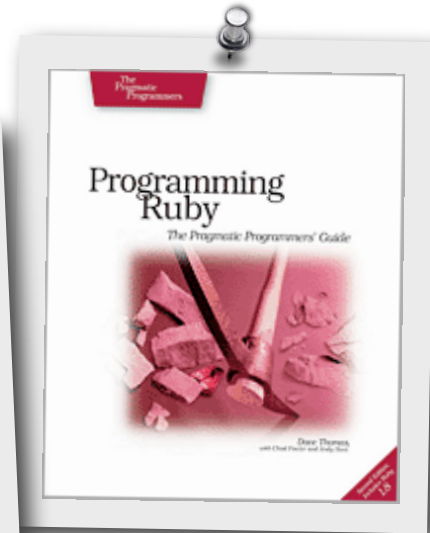
RUBY'S BACKGROUND

Created originally in an unlikely place—Japan—in 1995 by Yukihiro Matsumoto (known as “Matz” in the community), Ruby became international in scope when the first publications appeared in English in late 2000/early 2001. The first and most popular book, *Programming Ruby* by the Pragmatic Programmers, was

RUBY DEFINED

Ruby is a dynamic, general-purpose object-oriented programming language that combines many of the best functions of languages that have gone before. It was designed to be a programmer’s friend and to be a joy to

Photo © Dan Benjamin



The “Pick Axe” Book

widely known as the “pick axe” book because of the cover illustration.

Soon after, in early 2001, the first international Ruby conference was held in Tampa, Florida, with 30 people in attendance. After this meeting, David A. Black, Chad Fowler, and Dave Thomas (later replaced by Rich Kilmer), formed the non-profit Ruby Central, Inc., with the mission of promoting and expanding the Ruby language.

Ruby Central took on the task of operating international conferences about the language and underwriting regional conferences. Later, Ruby Central took over sponsorship of [RubyForge](#), a source site for Open Source Ruby projects originally established by InfoEther (author of this paper) to further community support efforts.

The language began to develop a significant following of developers through the efforts of many thought-leaders.

use, thus avoiding the love-hate relationship that software developers have had with other languages. Its creator stressed that its design emphasize human, rather than computer, needs.

It is relatively easy to learn, read, and maintain. It is less verbose and more code efficient than its predecessors, some say on a factor of 10-to-1 fewer lines of code (or more). Fewer lines of code meant fewer potential errors, which meant faster deployment and less support costs.

Ruby included some very advanced features such as meta-programming. Most of the early converts to the language were Java programmers who appreciated the efficiencies of Ruby and its speed and relative clarity over Java and other languages with which they were familiar.

THE TIPPING POINT—RAILS

However, the event that put Ruby over the top was the introduction of Ruby on Rails (normally just called *Rails*) in 2004 at the fourth Ruby Conference held in the Washington, DC, suburb of Chantilly, Virginia.

Introduced by a Danish programmer, David Heinemeier Hansson (popularly known as “DHH”), Rails was an Open Source “framework” created out of a for-profit collaborative product called *Basecamp*, which is operated by the company 37signals in Chicago. DHH opened the project up to a core team of international Open Source developers, which expanded Rails significantly.

Most major software applications today are developed to run on the Web, including those that run in so-called “cloud” computing environments (like Amazon’s EC2 or in “private clouds”—corporate server networks behind a firewall). At the core of such applications is a database.

Rails is a framework for developing these “database-driven” Web applications. As a framework, it has rules and conventions that provide a very rapid way to construct and deploy complex Web applications, typically in one-third to one-tenth the time and expense as an effort done with other alternatives. Once built, these sites are easier to maintain, extend, and evolve.

Rails used a [Model-View-Controller](#) (MVC) architectural pattern, which was not new. What was new, and what made it really different, was using it in combination with the concept of [Convention over Configuration](#).

Rails takes an “opinionated” approach. Such rules at first seemed constricting to programmers used to infusing their own personality into code, but proved to be more viable from a business standpoint because these opinionated rules made for more consistent standards and supportable implementations.

Additionally, Rails tied easily into the rise in Agile programming methodologies. Agile processes use short iterations by teams of programmers and stakeholders to develop complex software applications quickly that could adjust to constantly changing requirements. The Rails



David Heinemeier Hansson (“DHH”): Original Creator of Ruby on Rails

Photo © Wired Magazine

“convention over configuration” approach was well suited to Agile development.

Because of this, Rails found popularity quickly with investor-funded Web 2.0 start-ups since they could prototype and deploy their Web applications rapidly—which meant more efficient use of investor money.

CONFUSING RUBY WITH RAILS

As Rails achieved greater awareness, confusion ensued since the “Ruby on” part in the title was truncated in general discussion forums.

Rails was not a new language, but it was discussed in the popular technical press as though it was. Lost was the fact that as a framework, it was merely a collection of techniques written in the Ruby language to accomplish a specific purpose. For Rails to run, it needs a Ruby *runtime*—one of the versions of the language (see page 11 for an explanation of runtimes).

AN ODD MILESTONE

However, the popularity of Rails often obscures the fact it is written on top of Ruby, and this has introduced the confusion. Some less technical managers, who do not listen to their own developers, forgo understanding the relationship of the language with the framework.

This situation itself marks sort of a milestone. It illustrates that corporations, with their large IT staffs, are now turning to the Ruby/Rails solution where before it was the province of independent consultants, hobbyists, and hackers.

Training companies have seen this confusion manifested firsthand. For Rails developers to be truly proficient, they need to at least start to understand and program in Ruby to extend Rails where necessary in complex projects. Some managers who control the training budgets think Rails is its own language and thus veto classes in Ruby. To compensate for this ignorance, many Ruby courses are being relabeled by using Rails in the course title.

SPONSORS ENTER

Large technology companies also took notice of Ruby. Several are involved in creating optimized versions of Ruby with their own runtimes. These efforts are being pursued because these special runtimes will take advantage of programs already in place or in development sponsored by these companies.

Sun Microsystems developed the JRuby runtime to allow it to more easily run the vast number of Java applications and libraries available.

Microsoft is in the early stages of an IronRuby runtime to optimize .NET applications.

One of the most important milestones, however, was the decision by Apple in October 2007 to begin shipping Rails along with the main Ruby interpreter in the Mac OS X 10.5 operating system (known as *Leopard*) that comes with all Apple computers. Further, Apple is working on MacRuby, a Ruby

runtime optimized to its OS X operating system.

Several other companies are also creating optimized Ruby runtimes that seek to improve speed or processing or other functions (see page 13 for the list).

RUBY'S POPULARITY WITH PROGRAMMERS: **DISTILLED**

Ruby is an Excellent Language in which to Encode Domain-Specific Logic:

Ruby's flexible syntax makes the language virtually disappear, so what is left are domain terms. A "domain" reflects business or project specific knowledge—how to operate a travel site or process sales leads or the logistics required in refueling aircraft, are some examples. Once the domain is encoded in this Domain Specific Language (DSL) format, it is very easy for the domain experts to use that Ruby DSL. These domain experts may not be able to write it, but they can read and validate it. When there is talk of a 10-times developer productivity increase, this is the primary reason for it—the close communications between domain experts and those building the system.

Programmers Can Quickly Learn Ruby & Be Effective In It:

This is especially true if they learn it in the context of something they know (e.g., their domain). People actually enjoy writing the code, which is motivating for them to become more proficient in Ruby. Any good programmer can become a good programmer in Ruby.

Ruby Scales:

Contrary to some popular press reports that knock the Ruby on Rails framework for not scaling, that is, not expanding to be capable of handling large volumes of users/activities, the underlying cause of that impression was a misinterpretation of the facts. The case in point (Twitter) had a scaling problem that resulted from a software architecture design issue. Several projects on record have shown that Ruby can be pushed to levels that exceed all expectations because Ruby allows C code to be dropped into applications to perform computationally intensive operations. The ease of doing this is another key aspect of Ruby's scalability. Ruby can also be combined to use arcane languages like Erlang or Scala that deal with massive communications requirements.

BIG-TECH CORPORATE INITIATIVES WITH RUBY

Sun Microsystems: Created JRuby—a Ruby runtime to optimize Java-based applications (note: project migrated to Engine Yard due to pending Oracle acquisition)

Microsoft: Early-stage project to create a Ruby runtime optimized to .NET called IronRuby

Apple: Distributes Ruby within OS X (Leopard) on all Mac computers; created MacRuby—a Ruby runtime optimized to the OS X operating system

IBM: Released a Ruby driver & a Rails adapter for IBM databases including DB2

SAP: Released Blue Ruby a exploratory research project by SAP Labs in Palo Alto & Shanghai to create an enterprise-ready dynamic language environment for programming Ruby that runs inside SAP's ABAP Virtual Machine

“Ruby on Rails will replace Java as the programming language and platform with dominant market share. Java is vulnerable because alternatives such as PHP have proven viable for application development with dramatically lower costs.”

~ Christian Sepulveda, Pivotal Labs

REPLACING JAVA IN MIND SHARE & MARKET SHARE?

In March 2008, the [Gartner Group](#) published a report that predicted that the worldwide population of Ruby software developers would grow from less than a million to over four million by 2013, a 400% increase in five years.

Though the report was thin on details on how they arrived at that conclusion, it can be interpreted by looking at the current crop of new programmers coming to the language. Almost all have some fluency in Java. The report writers probably concluded that the worldwide population of Java developers would substantially shift to also doing or focusing more on Ruby (and Rails).

In 2003, Sun Microsystems, the creator of Java, was attempting to create a population of over 10 million developers, particularly emphasizing major outsourcing markets such as India and China, though subsequent reports guessed that it reached half that [target](#). Still, 5 million is a large number.

Java is vulnerable to being overtaken because Ruby, PHP, Python, and other languages have proven significantly viable as cost-efficient alternatives. When you include the power of frameworks like Rails in the mix, then it is understandable that many technologists and developers are seeing Ruby as the heir apparent to Java. Programmers are also always looking to increase their skill set and earnings potential, and those are powerful reasons to learn Ruby and Rails.

OR IS IT THE END OF A DOMINANT LANGUAGE ERA?

As part of this debate among programmers, some have noted that the only reason Java became dominant at the time it did was because the Internet and World Wide Web were immature. Now that they are more mature, the contention is that several languages, all Open Source, will share dominance and be selected and used based on the strengths of their supporting and contributing communities, which translates to their technical usefulness.

It also depends on the problems to be solved. Savvy developers will select the best tools for the job, and that may include several languages or components.

REALITY

The fact of the matter is that Java, currently with the dominant market share in the enterprise, *takes too long and costs too much in both development time and opportunity cost*. It is for these reasons that languages like Ruby and frameworks like Rails will eventually surpass it in the future.

Enterprise User Stories



Photo © James Duncan Davidson

SHADOWLANDS

Businesspeople and investors reading this paper are likely looking for ideas and examples as well as an education and a reference source.

It is no secret that most of the public discussion around Ruby and Rails has centered on start-ups, which have used them to create high-profile Web applications quickly in order to prove value for the investor cash they have received.

But one of the main trends this paper addresses is the move by enterprises—companies with traditional IT staffs—beginning their movement to Open Source technologies, particularly Ruby and Rails.

There is a dearth of publicity on this topic or available case studies of the use of the technology in such traditional organizations.

Most of the information that is available is anecdotal and not public because, at this stage, Ruby and Rails are being applied in many enterprises experimentally or for competitive advantage, and it is a little early to trumpet successes.

These projects are being developed in the shadows. Many deal with rather boring but vital reinvigoration of

internal processes that do not have the pizzazz of a Web 2.0 start-up, but they represent the next growth phase.

Enterprises—to use a loose definition—are companies or organizations of anywhere from 100 to 100,000 people. The traditional “small business” market portion of this group is usually entities with 100 to 2,000 people. Departmental groups within bigger organizations also fall into this category since they often behave and function independently.

Open Source technologies offer enterprises cost savings with lower “total cost of ownership” and a release from the lock-in of traditional proprietary vendors.

COMMERCIALIZATION PHASE

What needs to happen for Open Source technologies to take off in the Enterprise is for them to be more commercialized. That is, they need to be packaged up and simplified. An example of how this was accomplished was with [Red Hat](#), which commercialized the Open Source Linux operating system. This evolution is beginning to take place in the Ruby and Rails Ecosystem.

REPRESENTATIVE LIST

The authors of this paper have gathered many examples of how Ruby and Rails are being applied, both by start-ups and enterprises. Through firsthand project experience, interviews, and published reports, it is possible to provide the short list of projects that appears in this section.

For various reasons it is not possible to reveal the identities of the groups, companies, or organizations involved. Therefore, this list is offered purely to give the uninitiated reader a sense of the scope of the effort underway and some bottom-line results.

No attempt has been made to verify the accuracy of each example. The goal is to show the number of creative approaches being taken and to spark the reader’s interest.

| USER & TREND | DISCUSSION & SIGNIFICANCE |
|---|---|
| <p>Public Company—“Wrapping” Internal Applications</p> | <p>A CIO and his technical team use Rails to rapidly “wrap” or develop applications that pull data from monolithic corporate systems like SAP and others.</p> <p>This allows very quick response to business requirements without the massive expense of trying to develop in those systems—which sometimes cannot even be done.</p> <p>This was particularly important in meeting governance requirements (Sarbanes-Oxley, HIPPA, and other laws).</p> |
| <p>Government Agency & Bank— Replacing Lotus Notes and ColdFusion Applications</p> | <p>Each entity, separated by half a world, is replacing thousands of custom-built databases and applications written in those products in order to be freed from these proprietary applications and related higher total cost of ownership.</p> <p>Additionally, the Open Source aspect leads to greater flexibility and more future upside innovation.</p> |
| <p>Mobile Application Start-Up— Beating Off-Shore Development in Time and Cost</p> | <p>In formal competition to partially outsource the development of its ground-breaking mobile application, this start-up found a US-based Rails consulting firm was able to beat three major off-shore development firms in cost and development time—by a factor of three times less cost and four times less time.</p> <p>Off-shoring firms, while only 1/10th the hourly cost per consultant, were still using old technology (Java) that did not lend itself to an Agile and quick process development. Each off-shore entity required more than a year and a massive team compared to the US firm that delivered in less than three months at a third of the price—all by using Rails and a much smaller, more experienced team.</p> |
| <p>Major Financial Company— Reducing Internal User Support Calls</p> | <p>This international firm had 100 internal Web applications with 20% written in Rails and 80% in Java. Of the 1,800+ support calls received per month, only six concerned the Rails applications.</p> <p>The balance required a support staff of 40 to handle all the calls for the Java applications.</p> |
| <p>Multimedia Company— Replacing Unworkable Custom Applications</p> | <p>This company had an external team of 10 developers working 18 months to build a complex video-serving application using Java, which proved problematic.</p> <p>A Rails consulting company came in and re-created the same application in Rails with only three developers in six weeks. Rails deployment worked.</p> <p>Unworkable Java application and its team, dropped.</p> |
| <p>Research & Development Agency —Creating Rapid Prototypes</p> | <p>Ruby was used to develop prototypes to perform complex systems testing by a large distributed community of programmers. Prior to this effort, such a system had been believed to be unachievable by the staff.</p> <p>The ultimate test harness system built from this prototyping effort turned out to be one of the major accomplishments of the entire program.</p> |

Ruby Runtimes



DEFINITION & NEED

A Ruby runtime is a program that accepts Ruby source files, converts them into executable form, and then provides an environment in which they can execute.

So to run a program written in Ruby, including Rails, you need one of the Ruby runtimes that exists or is in the works.

With one exception, these runtimes are Open Source. They are effectively free and are downloadable from a number of sources. For the exception (MagLev), the source code may continue to be freely available.

In some cases runtimes may be packaged with a software application as part of the “plumbing” that makes it work, unseen by the average user.

Software developers need to use the best runtime for the application they are creating. Each of the runtime projects discussed in this section have their own special advantages.

COOPERATION

All of the various Ruby runtime groups work together in a mutually beneficial relationship. They share a common specification code-base, which they collaboratively evolve as a means of ensuring that their runtimes are compatible with the official MRI implementation (see below). In some cases, notably Rubinius and MagLev, they also share implementation code as seen in MagLev’s use of some of Rubinius’ standard library implementation.

The functioning of the entire Ruby (and Rails) Ecosystem depends on the collegiate and generous work done by these teams and a host of independent and dedicated developers around the world.

MATZ’S RUBY INTERPRETER (MRI)

The creator of Ruby, Yukihiro “Matz” Matsumoto, and a group of developers, mostly operating in Japan, currently develop the de facto Ruby runtime, called the “Matz’s Ruby Interpreter” or “MRI.” The current standard and most stable version is 1.8.7. A newer version (1.9) is available, and it represents a major rework that is significantly faster than the 1.8x branch. That makes 1.9 the fastest runtime on Ruby currently. A 2.0 version is also in the planning stages.

Each new version or update to the MRI includes new features, capabilities, and speed improvements and is closely monitored and used by the Ecosystem’s programmers.



BIG TECH COMPANIES MOVE IN

Major technology companies see advantages in supporting Ruby. Apple, Microsoft, and Sun Microsystems (which is soon to be purchased by Oracle) have all set up Open Source projects that have, or will, create their own runtime versions of Ruby.

These alternative implementations of Ruby generally leverage software and language features of products sold by each company. Basically they seek to optimize and extend software libraries or operating system features already available.

JRuby makes use of the vast number of Java libraries that exist. Microsoft will have an opportunity to extend its .NET development framework with IronRuby. Apple's MacRuby implementation will seek to expand the development of applications written for its OS X operating system.

In each of these instances, the companies have thrown people and resources behind these Open Source projects. Because they are all Open Source, they are available under licenses that are free to use.

Several of these implementations also seek to improve the speed of Ruby.



[JRUBY](#) is a Java implementation of Ruby developed as an Open Source project supported by [Sun Microsystems](#), the company that created the Java programming language. The project was originally created in 2001 and has supported Rails since 2006. In August 2009, the JRUBY team moved from Sun to the venture-funded [Engine Yard](#).



[IronRuby](#) is an upcoming Open Source implementation of Ruby targeted to the [Microsoft](#) .NET Framework. It is implemented on top of the Dynamic Language Runtime. Originally announced in 2007, the first Alpha version was released in late 2008. The IronRuby Microsoft team has also been working to support Rails, which they demonstrated in November 2008.



[MacRuby](#) is a version of Ruby 1.9 (latest), ported to run directly on top of the [Apple](#) Objective C runtime, which is the same engine that powers most of the Mac OS X operating system. The goal of this Open Source project is to enable the creation of full-fledged Mac OS X applications that do not sacrifice performance. [HotCocoa](#) is a UI tool that simplifies working with the OS X programming interfaces.



[MagLev](#) is a project by [Gemstone Systems](#), an enterprise software company based in Beaverton, Oregon, with offices in Asia, Europe, and India.

The goal of the project is to create "...a fast, stable Ruby implementation with integrated object persistence and distributed shared cache." It uses the company's mature and proven GemFire object database.

MagLev is in private Alpha testing with a Beta scheduled for 2009.

Rubinius

Ruby, the Way It Was Meant To Be

[Rubinius](#) is a virtual machine for running Ruby programs and a Ruby core library. It is an Open Source project sponsored by the venture-funded company [Engine Yard](#) (San Francisco, California).

Announced by Evan Phoenix in 2006, the work has involved capturing the specifications on the actual behavior of the MRI implementation. It has as its goal to make a more easily maintainable, modular, and hackable Ruby interpreter, making it more attractive to more programmers.

ALTERNATIVE RUNTIMES

In addition to the big players, smaller groups and companies are also involved in optimizing runtime versions of Ruby.

The two called out above, MagLev from Gemstone Systems and Rubinius by Engine Yard, were established with very distinct goals. The Rubinius effort got involved in the effort to document the specifications of Ruby itself. This led to a stand-alone project (<http://rubyspec.org/>) in which all the other runtime implementations are now collaborating.

As in the runtime project versions being developed by Microsoft, Apple, and Sun, these versions also seek to find ways to improve Ruby with an eye to increasing the number of software developers working in the language.

Some approach this by improving or adding performance. Others seek to make the language more accessible to allow more programmers to modify and extend their applications.

ACTIVITIES

A large community of companies and individuals are involved in creating alternative runtime versions of the Ruby language that improve or optimize it.

These groups willingly work together for the greater good of the community. An ultimate goal of all of these efforts is to expand the number of software developers programming in Ruby.

TRENDS

Most of the alternative Ruby runtimes are just now getting into the Alpha stage and are not yet in a position to have their impact felt.

During 2009 and 2010, however, this will change as these versions become more stable and capable. When they do, they will significantly contribute to the number of applications developed in Ruby.

OPPORTUNITIES

✓ All the companies supporting these projects stand to gain influence and market share for their core offerings based on the success of their Ruby runtime versions.

✓ A host of products and services stand to be built around each of the various runtime versions. Many companies in the Ecosystem are now positioning for just that situation.

Ruby Frameworks

BACKGROUND

A software framework is a conceptual structure that is used to solve a problem or allow a reusable design. It can support programs, code libraries, and other software used to assemble a solution.

After a point in time, most Open Source software languages have a large number of frameworks developed for them. Some find a niche, others get marginalized or fall into disuse. The ones that are successful tend to be so as a result of sufficient people, groups, and companies in a language's ecosystem building up value around them.

The most successful and notable Ruby software framework has been Rails, which has already been discussed in this paper in some detail. However, there are quite a number of other frameworks developed in Ruby that can be highly useful.

WEB APPLICATION DEVELOPMENT

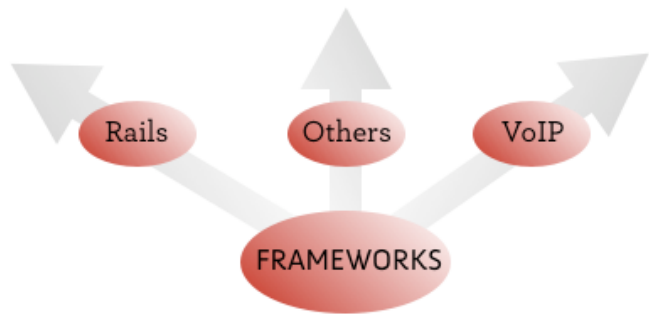
The graphic on this page from the Ecosystem graphic on page 4 shows a three-tiered Frameworks branch. Actually there are only two branches, one dealing with Web application development and one dealing with VoIP (Voice over Internet Protocol).

Since Rails has been so dominant, it is broken out as its own branch. The "Others" category includes all the other Web application development frameworks.

THE "OTHERS"

Lists (20+) can be found that assemble various Ruby Web application frameworks with a standard Web search. However, only a few have sufficient mind share, and many are marginal. Those few that matter and are in fairly wide use are listed as follows.

(Note that Merb is not listed here since it and Rails have merged and will appear as Rails 3.0.)



Rack: <http://rack.rubyforge.org/>

Rack provides a minimal interface between Web servers supporting Ruby and Ruby frameworks. It allows an extremely easy way to combine Web applications and has achieved a great deal of popularity in the developer community upon which other frameworks are sometimes built. It has its own core team of contributing developers.

Sinatra: <http://www.sinatrarb.com>

A Domain Specific Language (DSL) for quickly creating Web applications in Ruby. (A DSL is a specification dedicated to a particular problem domain, problem presentation, or solution technique.) It runs on Rack (as above). Unlike Rails, Sinatra does not use a Model-View-Controller (MVC) framework. It is very simple and powerful, and its code size is small, deploying as a single file. Because of these features, it has achieved a popularity in creating very fast, simple Web applications that do not have all the project requirements normally associated with a Rails implementation.

Ramaze: <http://ramaze.net/>

A modular Open Source Web framework that is designed to create a lighter-weight approach to Web application development and is an acknowledged alternative to Rails. It has developed a following that is supported by a core team of its own. In many cases Rails includes bundled functionality that does not get used. Ramaze is an approach that does not tie the developer down with too many unused bundled functions.

VoIP Framework: Adhearsion

This framework is significant in that it is different. It has the potential of becoming the next important Ruby framework beyond Rails but in an entirely new arena and has been characterized by some as the next Ruby “killer app.”

VoIP or “Voice over IP (Internet Protocol)” has been around from almost the beginning of the Internet. It effectively allows you to make voice phone calls via the Internet and even out to standard mobile or landline phones for minimal expense.

VoIP services include those offered to the home market and Skype, which is the hugely popular service that permits both voice and video conferencing over the Internet at very little or no cost depending on whether it goes outside the Internet to a landline or mobile network.

Adhearsion is an Open Source framework that is a new way to write (code) voice-enabled applications. The name comes from the combination of “adhesion” and “hear” because it integrates various Web 2.0 technologies with the “previously untamed telephony world.”

So how would this be used? Some examples include:

* Security Checkpoint: An application that combines a voice recording done at a security checkpoint with a

The logo for Adhearsion, featuring a stylized telephone handset icon to the left of the word "Adhearsion" in a serif font.

back-end database Web application over the Internet that would make a comparison for identification purposes to a remote secure server.

* Smart Corporate Phone Systems: Tie a VoIP office phone system into a corporate Web application to develop or link into billing systems, track calls and compare them to elements in a database, or select dial plans, etc.

* Parking Space Availability: Use Intelligent Voice Recognition (IVR) to record the availability of municipal parking spaces that could be broadcast to mobile devices.

* Phone Surveys: Relate a call survey to a custom-built Rails application.

Adhearsion is an Open Source project created by Jay Phillips. Adhearsion is also the name of Jay’s company that provides telephony consulting.

Adhearsion announced a relationship with Voxeo to launch Voxeo Labs to create a “hosted telephony engine.”



New Framework? Ruby on Mobile Devices

Though not specifically related to VoIP, there is activity to permit the use of Ruby to create applications or frameworks for applications on mobile devices. One Open Source offering is called Rhodes from Rhomobile, which has the goal of creating a Ruby framework on smart phones. Another initiative called the Mobile Ruby Project is an effort to create a Ruby virtual machine for the Symbian smart phone operating system, which is used by a number of handset vendors.

Community Support

BACKGROUND

Beyond Ruby implementations and frameworks, the largest number of players in the Ecosystem can be found in four major subgroups, each with one or more segments. The first to be dealt with here are those that operate in what is best called the "Community Support" arena.

Community Support includes training, publishing, and conferences as depicted in the graphic. These functions appeared first and helped to spread the initial interest and remain as the main form of promotion and learning in the Ecosystem.

UTILIZATION & INVESTMENT

The first place most software developers discover Ruby and Rails is through technical books and publications. This is usually augmented by studying blogs and online articles.

Initial reading and research is usually followed by attending a conference, whether national or regional. At conferences one gets more detailed interaction on specific topics from both the presenters and the fellow attendees.



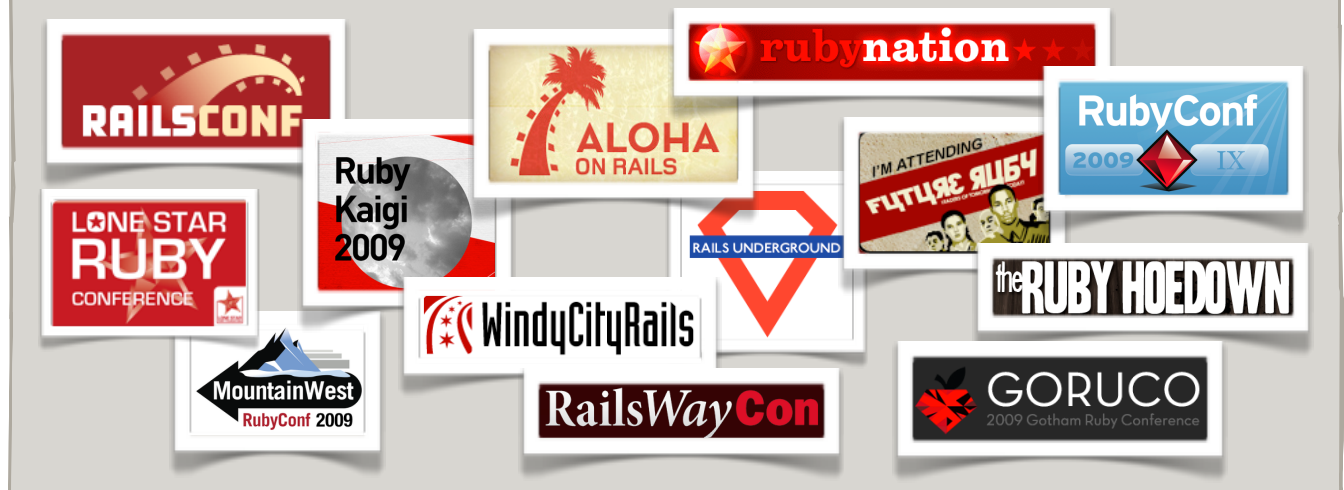
The next step generally involves more formalized training that is offered by a number of groups and consultants. Such training is normally in a classroom setting lasting several days with workshops and course materials.

Formal training is particularly popular with Enterprises and organizations wishing to get entire teams up and running on the new technologies and approaches.

All of the segments discussed in the following pages under "Community Support" usually see investment coming from individuals and companies in the form of self-investment or sustained through operating revenues.

This is one of the most vibrant and active areas of the Ecosystem and represents its public face.

Sample Ecosystem Conference Logos



Community Support: Conferences



Photo © James Duncan Davidson

SEGMENT BACKGROUND & HISTORY

Perhaps the most productive learning environment for software developers are conferences built around languages and frameworks. The energy at these events is palpable.

Conferences held around Ruby and Rails have developed into the main events around which the Ecosystem of the community has been built. They provide forums for the introduction of advances created by community members. They serve to train and teach. And, most importantly, they allow face-to-face interaction and classic “networking.”

In most cases at the bigger events they turn into “codefests” where groups of programmers get together for ad hoc

programming sessions to solve problems or develop new techniques. In most cases these individuals only interact online until these events.

For example, at one of the early Ruby events, the packaging technique called Ruby Gems was produced during several late-night sessions. “Gems” became critical to the entire deployment capabilities of Ruby applications, including Rails.

REGIONAL PROLIFERATION

The first international Ruby conference was held in 2001 in Tampa, Florida, attended by just 30 people. Out of that grew Ruby Central, Inc., a non-profit that was set up to promote the language through annual conferences, and other means. Initially Ruby Central also underwrote several local conferences but these have now grown to the stage they are self-sustaining. While the two major annual events, RubyConf and RailsConf still draw the largest crowds (500-2,000 attendees), the numerous regional conferences are now the predominant

PLAYERS

Ruby Central

O'Reilly Media

Regional Groups

KEY POINTS

- ★ International, national, and regional conferences are the key events that facilitate the growth of the Ruby & Rails Ecosystem
- ★ These events serve to introduce, inform, train, exchange, and produce advances in the community
- ★ Nearly 100 local groups have been established around the world, some of which put on local conferences but also meet frequently with topics of interest. A complete list of these users groups can be found at <http://ruby.meetup.com/>

events for most people in the community.

These regional events are usually national in scope though some are attaining international coverage as well.

ACTIVITIES

Two major events dominate the Ecosystem—RubyConf, which focuses on the Ruby language, usually held in the fall, and RailsConf, which focuses on the Rails framework, usually held in the spring.

Both are put on by the non-profit Ruby Central. RailsConf is the larger event and is done in partnership with [O'Reilly Media](#).

TRENDS

Besides the two major events, there are numerous significant regional events that can cover several states or several countries. These regional events can attract several hundred attendees.

More regional and local events are appearing each year, and many have now taken a focus on more granular topics of interest. They are the most cost-effective conferences for most developers to attend.

OPPORTUNITIES

- ✓ No real investment opportunities exist in this segment.
- ✓ There are opportunities for local groups to realize a profit from putting on regional events, which also serve to advance the Ecosystem.
- ✓ There is plenty of room for more focused events around subset community topics (e.g., MacRuby, etc.).

Community Support: Conferences

| MAJOR INTERNATIONAL CONFERENCES | |
|--|--|
| RubyConf http://www.rubyconf.org | This is the premier event for the Ruby language community and is run by Ruby Central itself in the late fall. It has been operating since 2001 and is usually moved to a different city in the US each year. Attendance is typically limited to less than 500. |
| RailsConf http://en.oreilly.com/rails2009 | The largest event in the Ruby & Rails Ecosystem. It is put on as a partnership between Ruby Central and O'Reilly Media. In 2009, the fourth such conference was held. Attendance was down due to the world economic downturn but was still substantial with 1,300 attendees. This event also has a large number of sponsors that display in an exhibit area. |
| RailsConf Europe http://en.oreilly.com/railseurope2008/public/content/home | This event was also held as a partnership between Ruby Central and O'Reilly Media. It was held for several years, usually in Berlin, Germany, in the early fall, but was suspended in 2009 due to the economic downturn. Several regional conferences in Europe stepped up to fill the void (see below). |
| SIGNIFICANT REGIONAL CONFERENCES (PARTIAL LIST—ALSO CONTINUED ON NEXT PAGE) | |
| RubyKaigi http://rubykaigi.org/2009/en | Technically this is an international event but as it is held in Japan with the majority of presentations in Japanese, it is classified here as a regional conference. Many keynote speeches and some work sessions are held in English. |
| RubyNation http://rubynation.org | An annual Ruby conference serving the Virginia, West Virginia, Maryland, and Washington, DC, areas. The mid-Atlantic area is a major technology corridor due to the proximity of the federal government where numerous Ruby and Rails projects are being pursued. |
| Scotland on Rails http://scotlandonrails.com | A highly rated small conference (>200) held in Edinburgh, Scotland, for UK and European attendees. It is one of the events that filled the gap left by RailsConf Europe. Draws sponsorships and US keynote speakers. |
| Lone Star Ruby Conference http://www.lonestarrubyconf.com | Held in Austin, Texas, for several years, it draws developers from the across the region and well-known speakers including "Matz," the creator of Ruby. It has become one of the best-known and attended regional conferences in the Ecosystem. |
| GoGaRuCo http://gogaruco.com | Short for "Golden Gate Ruby Conference," this conference is held in San Francisco and is the major local Ruby event (>200) to address the Silicon Valley region. Ruby and Rails are the technologies of choice for many Web-based start-ups in the region. |
| RailsWayCon http://it-republik.de/konferenzen/railswaycon/ | Another of the events to fill the void of RailsConfEurope, this conference is held in Berlin, Germany, and draws Rails developers from across Europe. The 2009 event drew international speakers and a large number of sponsors. |
| Aloha on Rails http://alohaonrails.com | A new regional conference to be held in Hawaii that will draw international attendees. Rails is used extensively in Asia, Australia, and New Zealand and will also draw from the US West Coast. |

Community Support: Conferences

SIGNIFICANT REGIONAL CONFERENCES (PARTIAL LIST—CONTINUED FROM PREVIOUS PAGE)

| | |
|--|--|
| Rails Underground http://www.rails-underground.com/ | <p>Another new European event held in response to the suspension of RailsConf Europe. It was the first major Rails conference set in London and drew both international speakers and sponsors in 2009.</p> |
| EuRuKo http://www.euruko2009.org | <p>One of the largest (>300) and oldest European Ruby conferences that has received quite a following for the past few years. It was held in Barcelona, Spain, in 2009 and Prague, Czech Republic, in 2008. In its first years it was held in Munich, Germany, and Vienna, Austria.</p> |
| MountainWest RubyConf http://mtnwestrubyconf.org | <p>Held for three years in Salt Lake City, Utah, this was one of the first successful regional conferences. It focuses on working sessions and “hackfests” that dealt in more hands-on development.</p> |
| Ruby on OS X http://rubyonosx.com | <p>An example of a small regional conference that focuses on a subset of Ecosystem topics, in this case Ruby on OS X, the operating system for Apple computers. This event was held in Amsterdam, The Netherlands, and drew a good crowd in 2009 for its work sessions.</p> |
| RubyRX http://www.nfjsone.com/conference/raleigh/2009/02/index.html | <p>A regional event that is held on the US East Coast, alternating between Philadelphia, Pennsylvania, Raleigh, North Carolina, and other cities. It is focused on training and mentoring and addresses the use of other languages used with Ruby and Rails.</p> |
| GoRuCo http://www.goruco.com/ | <p>Short for Gotham Ruby Conference, this event is held in New York City for about 150 developers and is usually a one-day event. It is organized and run by volunteers from the New York area Ruby group.</p> |
| FutureRuby & RubyFringe http://futereruby.com | <p>FutureRuby grew out of past RubyFringe events and is an avant-garde conference for “Rubyists” that is held in Toronto, Canada. It is a single-track independent conference with the goal of driving innovation with no paid technical sponsors and a target attendance of 150.</p> |
| Ruby Hoedown http://www.rubyhoedown.com | <p>A Ruby conference focused on the Southeast US. In 2008 it was held in Huntsville, Alabama at the University of Alabama, and drew a large number of key speakers and sponsors.</p> |
| LA Ruby Conference http://www.larubyconf.com | <p>Another new conference held in Los Angeles, California, in April 2009 that attracted over 100 developers and a range of sponsors. This event also used the single-track format so participants could all share a common experience.</p> |
| WindyCityRails http://windycityrails.org | <p>A one-day Rails conference with tutorials and coding workshops held in Chicago, Illinois, by the local Ruby and Rails enthusiasts in September. The group was able to donate some of its proceeds in past meetings to charities. Draws a large number of sponsors.</p> |
| Rails Summit Latin America | <p>An ambitious attempt at giving South America its own large Rails event. The 2009 event is in the planning stages for a October conference to be held in Sao Paulo, Brazil. It will mark the second event in this series. (English-language link: http://www.railssummit.com.br/en/pages/home)</p> |

Community Support: Training



Photo © The Pragmatic Studio

...

SEGMENT BACKGROUND & HISTORY

The trend that follows the learning of new technology, particularly software, applies here as well.

Advances are so rapid that traditional learning institutions—universities, colleges, trade schools—are usually the last to adapt curriculum to teach new software languages.

The method that has evolved is one where people learn through a combination of online exchanges (Web sites, blogs, Twitter, etc.) followed by reading first versions of technical books, until the first formalized training courses appear.

In most cases, large traditional training companies are not the ones offering

such courses in groundbreaking technologies.

Small consulting-oriented companies and individuals take on the task first usually as an extension of activities of the authors and publishers in the field.

This is the case with Ruby and Rails. The landscape is dominated by early movers who offer limited and very technical courses of very high quality.

Typically the courses last several days to a week, involve 50 people or less, and usually require attendees to actually program in the language or framework being taught. Experienced instructors are used as a major differentiator by the players.

AN EXPANDING SEGMENT

Listed companies are not the only ones providing training. Consultancies are also offering training as an adjunct profit center or to the delivery and transfer of major projects to in-house tech teams.

PLAYERS

The Pragmatic Studio

RubyPal

Obtiva

Think Relevance

Marakana

Big Nerd Ranch

Webucator

KEY POINTS

- ★ Training needs have been consistent despite the economic downturn
- ★ The available talent pool of qualified trainers is strained as consulting offers greater income potential
- ★ Corporate & government agencies are becoming the major training customers displacing funded tech start-ups
- ★ Training courses, tech books, & conferences represent the principal means of learning Ruby & Rails

ACTIVITIES

The pace and size of the training activity in Ruby and Rails is requiring all the current players to adapt and expand.

The greatest issue with this expansion is the increasing lack of experienced and qualified instructors, most of whom are finding it far more lucrative to do consulting rather than teaching.

TRENDS

There is not yet any apparent consolidation activity, and no one player is emerging to dominate.

The likely trend in this segment is that more large-scale IT training companies will begin to offer Ruby and Rails courses, further straining the available training talent, which may lead to some acquisitions of current companies in this niche.

OPPORTUNITIES

- ✓ Consolidation of existing players or consolidation with large consulting companies is possible.
- ✓ Increasing requirements for experienced training professionals will offer opportunities for new players to enter this segment.
- ✓ Known, experienced instructors provide differentiation between players.

Community Support: **Training**

| TRAINING STUDIOS FOCUSING ON RUBY & RAILS | |
|--|--|
| The Pragmatic Studio http://pragmaticstudio.com | One of the best known in this segment. They are associated with The Pragmatic Programmers (http://pragprog.com), a technical book publishing group. The Pragmatic Studio has the reputation of providing top-level course materials and instruction offered by some of the best-known programmers in Ruby and Rails. This group also offers courses in associated technologies such as iPhone development due to the fact many Ruby and Rails students return for iPhone training as Ruby/Rails are often used as back-end web services for iPhone applications. The RESTful conventions used by Rails are making this integration easier. |
| CONSULTANTS & PUBLISHERS SPECIALIZING IN RUBY & RAILS | |
| RubyPal http://rubypal.com | Independent consultant and author David A. Black conducts training for groups and companies. He is one of the co-founders of Ruby Central, the non-profit set up to promote the Ruby language. |
| Obtiva http://obtiva.com | A small Chicago-based consulting group that does training and coaching in Agile software development including Ruby, Rails, J2EE, etc. |
| Think Relevance http://thinkrelevance.com | A North Carolina-based consulting and training company that also publishes a number of technical books. They specialize in project and security audits, testing, and architectural reviews. |
| TRAINING STUDIOS—GENERAL IT FOCUS | |
| Marakana http://marakana.com | A San Francisco-based, self-funded company, that targets onsite training for corporations and government. Topics include Java, Ruby, Agile development, Adobe, and Web development in general. This group also offers public sessions. |
| Big Nerd Ranch http://bignerdranch.com | A company that conducts training at retreats with labs (boot camps). Offices are located in Atlanta, Georgia, and Germany. Training is conducted in a broad range of programming languages and operating systems. |
| Webucator http://www.webucator.com | A more classic IT training shop with offices throughout the country offering an entire range of technical courses of which Ruby is a small portion. |

Community Support: Publishing



...

SEGMENT BACKGROUND & HISTORY

This is probably one of the most important segments to the growth of the Ecosystem. Books published on Ruby and Rails represent the main method for programmers to learn the technology.

The first major book credited with starting the international interest in Ruby, *Programming Ruby*, was known as the “pick axe” book because of its cover illustration. It was published in late 2000 and was the most important English-language document to introduce Ruby to a worldwide audience.

The number of publishers has now expanded, and many of the best-known personalities in the Ecosystem have published titles.

A MODERATION IN SALES

An annual “State of the Computer Book Market” posted on the [O’Reilly Radar blog](#) in February 2009 indicated the overall market for programming languages was down 5.9% in 2008 when compared to 2007. There were 1,849,974 units sold in 2007 versus 1,740,808 sold in 2008.

In 2007 Ruby grew very well passing Perl and Python but saw the largest decrease in unit sales in 2008. However, there was dramatic interest in Objective-C, the language used for development on Apple’s iPhone, and many Ruby programmers did shift to iPhone development in 2008.

Interest also stepped up in lighter-weight languages like PHP and ActionScript, both used extensively for in-browser development, which may signal more junior people or non-programmers are beginning to see the need to increase their learning to support the creation of simpler Web applications.

PLAYERS

Addison-Wesley

Apress

Manning

O’Reilly Media

Pragmatic Bookshelf

Sams Publishing

Wrox

KEY POINTS

- ★ Published computer books represent the most timely and effective method for distributing highly technical information
- ★ Purchased technical titles are generally the first method used by people to learn about a technology, regardless of the \$40-\$80 unit prices
- ★ There was an overall decline in technical book sales in 2008; it may be a function of the economic downturn or some other factor

ACTIVITIES

Any major publisher that has a history of focusing on technical books has already become involved in putting out titles on Ruby and Rails.

Many of the titles that have become popular are now in second or third editions.

Over 100 titles on Ruby and Rails are now in print.

TRENDS

Technical book publishing saw an overall drop in the total number of titles sold in 2008, but the drop in computer language titles was actually less than the overall downtrend.

Most of the major topics have been covered, but new segments continue to arise where new titles will be needed and forthcoming, as in MacRuby, IronRuby, etc.

OPPORTUNITIES

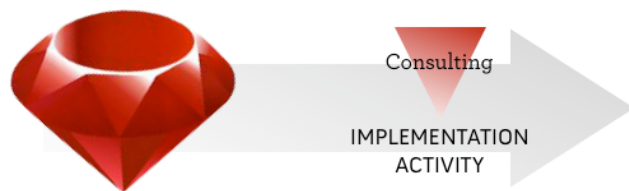
✓ Print publications may continue to decline or be replaced by electronic or downloaded versions, which will be less costly but not require the traditional distribution network; such innovation will prove a major differentiator.

✓ New entrants are unlikely in this segment, and further consolidation is possible.

Community Support: Publishing

| FOCUSED TECHNICAL BOOK PUBLISHERS | |
|---|---|
| Apress http://www.apress.com | Apress books are characterized by distinctive yellow and black cover designs. Most of the Apress titles are Microsoft-centric. Their books are distributed worldwide by Springer-Verlag, a worldwide publishing company based in Germany. |
| O'Reilly Media http://www.oreilly.com | Formerly known as O'Reilly & Associates, this Sebastopol, California-based media company publishes books and Web sites and produces conferences on computer technology topics. Founded by Tim O'Reilly, who was once described as the "guru of the participation age" and who is credited with coming up with the term "Web 2.0." O'Reilly books have a consistent shelf appearance with branding that features a woodcut or animal line drawing on the book covers. |
| Pragmatic Bookshelf http://www.pragprog.com | Characterized as an Agile publishing and training company, the "Prags" as they are known in the community publish some of the best-known Ruby and Rails books and perhaps have the most focus on these topics of all the publishers on this list. Founded and run by Andy Hunt and Dave Thomas, both are very well-known personalities in the Ecosystem and conduct training studios with the Pragmatic Studio (http://pragmaticstudio.com). |
| Manning http://www.manning.com | Centered in Greenwich, Connecticut, Manning focuses on computing titles at professional levels. Their books are aimed at programmers, systems administrators, designers, architects, and managers. |
| COMPUTER BOOK DIVISIONS OF GENERAL PUBLISHERS | |
| Addison-Wesley http://pearsonhighered.com | A book publishing division of Pearson PLC, it is best known for its computer book series that is distributed through the Safari Books Online e-reference service. |
| Sams Publishing http://www.sampublishing.ca | A subsidiary of Pearson Education that is dedicated to the publishing of books on computer programming. Notable for the Teach Yourself, "Unleashed," and "21 Days" book series. |
| Wrox http://www.wrox.com | A computer book publisher originally based in the UK, it was acquired and now operates as a division of John Wiley & Sons, Inc., of Hoboken, New Jersey. All books published by Wrox are written by software developers. |

Implementation Activity



BACKGROUND

This branch of the Ecosystem is probably the next most visible to the outside world after the Community Support branch. Most non-technical people first learn of the Ruby and Rails Ecosystem through outside consultants who are actually applying the technology to solve problems and meet business needs.

These companies are very visible evangelists to businesspeople because they know how to make the technology produce the dramatic productivity claims that would otherwise be dismissed as hype.

ESTABLISHED FIRMS

The sample representative consulting companies profiled on the following pages are pioneers of sorts. Most started with the express mission of pursuing Ruby and Rails projects on a commercial basis.

A few of them are international, and only one is large by consulting standards.

The major technology consulting players are late to the party. Like most typical large companies, they are busy protecting their business models which are built around older technologies that are on the wane.

INDEPENDENTS

Most of the profiled firms here are small businesses with 5 to 20 employees. By far, though, the largest segment involved in Ruby and Rails implementations are independent contractors—many hundreds of them. They provide the backbone for the range of work being done in both start-ups and enterprises with Ruby and Rails. These profiled firms rely heavily on subcontracting these independents.

HOSTING FIRMS AS IMPLEMENTORS

Several venture-funded companies that are profiled elsewhere in the “Implementation Tools” branch are also involved to some extent in implementation activities. However, even in those cases, either a consultant or a savvy in-house developer is involved.

Sample Ecosystem Consulting Firm Logos

Implementation: Consulting



Photo © James Duncan Davidson

...

SEGMENT BACKGROUND & HISTORY

Companies engaged in Ruby and particularly Rails consulting represent one of the most active segments in the Ruby & Rails Ecosystem.

Many of the consultancies that specialize in Ruby and Rails are small businesses with 5-20 people. Only one multinational company is on the list as it was one of the earliest supporters of the Ruby language and actively develops almost half of its projects using the Rails framework.

The majority of the consulting projects delivered to date have been to funded technology start-ups. However, this is changing, and more corporate-class customers are showing up.

The types of consulting being provided in the Ruby and Rails Ecosystem involve everything from coaching internal teams to code audits, best practices, and, of course, full-on coding of applications.

CONTRIBUTIONS & PRODUCTS

These consulting companies also are involved in contributing to Open Source projects, and most are involved in building software products that contribute to solving many implementation, configuration, or other challenges in the community.

Several of the players perform custom training sometimes in partnership with one of the companies on the Training Segment list (see page 20). Consulting projects are handed off to clients who use the training to bring their technical staffs up-to-date.

INDEPENDENTS

In addition to the companies listed, a large group of independent software developers perform consulting as subcontractors.

PLAYERS

ELC Technologies

ENTP

Hashrocket

Highgroove Studios

InfoEther

Intridea

Pivotal Labs

ThoughtWorks

KEY POINTS

- ★ Major implementations of Ruby & Rails commercial projects are being performed by an exclusive list of consultancies containing some of the best-known people in the Ecosystem
- ★ Projects continue to grow in complexity & function using Agile methodologies
- ★ Players in this segment are also taking the lead in developing products for the Ecosystem

ACTIVITIES

Players continue to add staff as their operations expand to meet demand, which continues to increase steadily despite the economic downturn.

The one large global company on this list (ThoughtWorks) has seen the number of its Ruby and Rails projects expand to account for 40% of its business. All on the list use Agile development processes for which Ruby and Rails are known.

TRENDS

The make-up of the customer base for these companies is gradually transitioning from technology start-ups to midsize business and enterprise-class corporate clients.

A few of these consultancies are creating products that target unmet needs or address repeatable processes they have seen in their customer projects.

OPPORTUNITIES

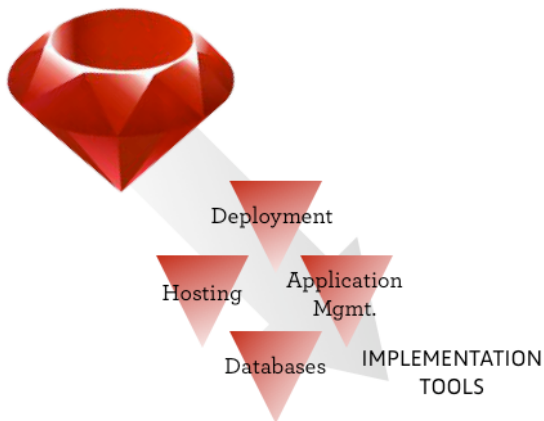
✓ Increased interest by large companies in Ruby and Rails projects will deny some of the current players entering that market because only a few contain consultants with Enterprise-level experience or mind-set. This will lead to a shake out and consolidation.

✓ Consultancies that do make the transition will grow rapidly.

Implementation: Consulting

| CONSULTING FIRMS: SMALL TO MID-SIZED | |
|--|---|
| ENTP http://entp.com | Consultancy and Web incubator collective based in Portland, Oregon. The company has been very active in creating and contributing to Open Source projects as well as doing innovative client work. |
| Hashrocket http://hashrocket.com | Based in Jacksonville Beach, Florida, and described as a boutique consultancy, the founder, Obie Fernandez, is an author, speaker, and well-known blogger in the Ruby space. Hashrocket ran its own learning and networking conference in August, 2009— BizConf . |
| Highgroove Studios http://highgroove.com | A consultancy and training company with offices in San Francisco, California, and Atlanta, Georgia. Team members are authors of several Ruby and Rails books. Highgroove has released a Rails instrumentation product for application performance monitoring called Scout. |
| InfoEther http://infoether.com | Founded in 2001, InfoEther was perhaps the first US company to make revenue from Ruby by applying it to several DARPA (US Defense Department's R&D agency) projects beginning in 2002, using the language to create a massive automated test environment. The company is developing a number of software products. Two of its executives, Rich Kilmer and Chad Fowler, are members of the non-profit Ruby Central, which has the goal of promoting and supporting the Ruby and Rails Ecosystem. [Note: InfoEther is the author of this white paper.] |
| Intridea http://intridea.com | Located in Washington, DC, Intridea provides software development consulting and is developing its own range of software products. Its staff is heavily involved in the Ruby and Rails community. |
| Pivotal Labs http://pivotalabs.com | Located in the San Francisco Bay Area, Pivotal has a strong reputation with many leading personalities in the Ruby, Rails, and Open Source communities. They achieved recent recognition on an engagement to help Twitter, the popular micro-blogging service. The company has also developed a number of project management tools and contributed to many Open Source projects/products. |
| CONSULTING FIRMS: MULTINATIONAL | |
| ELC Technologies http://elctech.com | Based in Santa Barbara, California, with an office in London, United Kingdom, ELC is the oldest firm in this group having been founded in 1991. It changed over to specializing in Rails development and has done work for some major clients including Cisco and ESPN as well as many funded technology start-ups. |
| ThoughtWorks http://www.thoughtworks.com | By far the largest entity in this category with over 1,000 employees in offices across the globe, ThoughtWorks has reported that over 40% of its projects involve Ruby and Rails. Headquartered in Chicago, it has offices in the UK, Continental Europe, India, and China. It fits the classic model of a large-scale consulting house with customers in financial, government, and media sectors. Its founder, Roy Singham, is a passionate advocate of Ruby and Rails, and his company has been a supporter of the Open Source community. Additionally, the company has spun off a product division as ThoughtWorks Labs (http://studios.thoughtworks.com), which offers Mingle, an Agile software development support tool. |

Implementation Tools



BACKGROUND

This is the largest single branch of the Ecosystem in terms of activity and investment. What is being developed here is critical to making the technology work—to make it viable to the end user.

IN THE CLOUDS

The vast majority of the external investment has been in companies in the “Hosting (Applications)” segment. Most of companies receiving investment have focused on delivering so-called “cloud-based” solutions. That is, they are providing an external data center through which Rails applications are deployed, run, and maintained.

PUBLIC OR PRIVATE?

Cloud computing is a hot investor topic and is relatively new. Enterprises see great cost savings potential in them, but they face huge security and governance challenges in using them for critical applications, particularly if they are public companies.

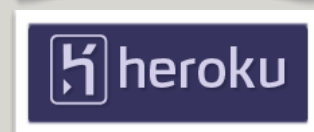
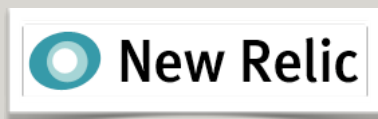
Putting your applications and data on data centers owned and run by others can be an issue and has given rise to the idea of “public” and “private” cloud-based computing.

“Private clouds” are run behind a firewall. Some hosting vendors are developing hosted options in this category, but many enterprises are planning to extend their current infrastructure investments to run their own. Enterprises have a strong bias in maintaining control.

OPPORTUNITIES?

There is an opportunity in figuring out how to run Rails applications behind the firewall within enterprises, particularly since enterprises have their own unique way of doing things—contrary to the way most start-ups currently operate.

Sample Logos of Firms, Sites, & Products



Implementation Tools: Databases

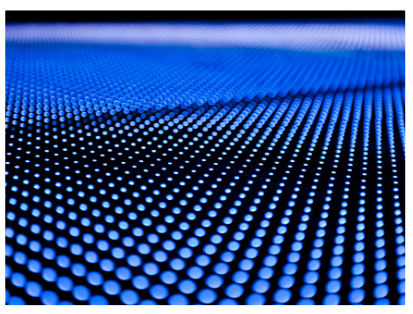


Photo © James Duncan Davidson

...

SEGMENT BACKGROUND & HISTORY

The reason for the existence of Rails is for the creation of Web-based applications that are driven by a database.

Hence, databases are critical to those using Rails. However, the field, for various reasons, is rather limited.

As Ruby and Rails are Open Source, it is no surprise that the virtual default database used in Rails by the majority of developers is the Open Source MySQL database. MySQL was purchased by Sun Microsystems in 2008. Sun Microsystems itself is being purchased by Oracle in 2009.

This would seem to limit the field, but that is because databases are actually a mature technology. There is room for

new entrants, and Rails can use whatever comes next.

THE PLAYERS

MySQL: <http://www.mysql.com>

This is the dominant and most popular Open Source database that is used by other languages/frameworks besides Rails. The company has been no real concerted effort to promote itself and is currently enjoying its status as the default choice.

EnterpriseDB: <http://www.enterprisedb.com>

This is the real challenger to MySQL in the Open Source arena with its PostgreSQL offering. The company was invested in by IBM and has been making strides to get both its Open Source and commercially licensed versions into the Rails community.

Oracle: <http://www.oracle.com>

Oracle databases are being applied to Rails primarily to use encryption techniques or to accommodate legacy systems. Because Oracle databases are proprietary and have a significant cost,

PLAYERS

Sun (MySQL)

EnterpriseDB (PostgreSQL)

Oracle

KEY POINTS

- ★ Rails is a framework for developing “database-driven” Web applications
- ★ Most of the databases used with Rails are Open Source (i.e., no license fees)
- ★ The dominant database in use with Rails is the Open Source database MySQL
- ★ Little innovation represented by new entrants in the database field has occurred
- ★ Current players are innovating within their existing offerings

they are only being used today in certain Enterprise clients, most of which have already paid for licenses.

ACTIVITIES

This is a segment with very little activity because of the default choices of either MySQL or PostgreSQL by developers as these are both Open Source.

Database vendors have not targeted the Ruby and Rails Ecosystem per se.

Open Source database vendors do continue to push aggressively in the larger Open Source world.

TRENDS

Consolidation within the larger database vendor space has limited options, and no new entrants have emerged of note.

Current vendors do continue to offer improvements and new features.

Legacy databases for which Enterprises have purchased licenses for other platforms are now beginning to be connected in Rails applications.

OPPORTUNITIES

✓ No real investment opportunities exist in this segment.

✓ There does continue to be opportunities for developers to create libraries and subset technologies that use current databases in unique ways.

✓ Current vendors do have the ability to innovate more with more refined and targeted offerings.

Implementation Tools: **Application Management**



...

SEGMENT BACKGROUND & HISTORY

Often referred to as “performance monitoring,” this service provides tools that monitor Ruby and Rails applications to show developers how their fielded software is performing technically. This is accomplished through various analysis and display techniques.

BRINGING AN ESTABLISHED SERVICE TO THE ECOSYSTEM

Performance monitoring firms have long been started and establishing in all the various software languages. So, it is no surprise that they have come to Ruby and Rails. Tools are developed and offered by the service that monitor an application, spotting potential problems and offering developers clues

to detect the causes, which is the first step in isolating and solving them.

There are just two companies that have been funded to focus on application management in the Ruby and Rails Ecosystem:

FiveRuns: <http://www.fiveruns.com>

This Austin, Texas-based firm funded by Austin Ventures was recently acquired by Workthink, Inc. (<http://www.workthink.net>).

NewRelic: <http://newrelic.com>

Based in San Francisco, California, the company was founded by veteran CEO Lew Cirne who had earlier founded Wily Technology, a company that became a market leader in Web application performance management for large enterprises. NewRelic was created to bring this expertise to the Rails community and was venture funded by Benchmark Capital and Trinity Ventures. Its main product is RPM, which has achieved a loyal

PLAYERS

FiveRuns

New Relic

See Also “Hosting (Applications)” Segment for Others

KEY POINTS

★ Application Management is the monitoring of the performance of a software application so that problems can be found and corrected and tuned

★ These types of services and tools have existed in other languages and have been brought to Ruby and Rails in optimizations

★ Such tools/services are necessary for the Ecosystem to function properly and to grow

following and allowed the company to set up an affiliate program.

ACTIVITIES

One of the first areas of venture capital investment in the Ruby and Rails space was in application performance management.

Tools offer 24x7 monitoring and present matters in a dashboard metaphor with charts and graphs.

Pricing typically involve monthly fees per host (server).

TRENDS

Tools in this segment continue to evolve and expand.

Some application of the tools are being expanded into other languages by one of the vendors.

These tools are primarily being used by start-ups and are just beginning to appear in Enterprise projects.

OPPORTUNITIES

✓ This segment seems to be maximized at the moment.

✓ Success or expansion of the segment depends on the continued growth of the Ecosystem and the uptake by corporations and enterprises.

✓ Combining these tools with other products and services will be more attractive to Enterprise customers.

Implementation Tools: **Deployment**



“Rails applications are easy to build, but sadly have been awkward to deploy.”

~ Martin Fowler, Chief Scientist, ThoughtWorks

SEGMENT BACKGROUND & HISTORY

Open Source technologies offer many advantages and savings, but they also present some challenges. As noted in the quotation above, the actual deployment of a Rails application requires some specific knowledge and dedication because there are numerous components required to make it work.

These components come from many sources. The graphic on the next page visualizes most of these pieces.

In addition to the Rails application that is actually written by a developer, for it to run you need a Ruby runtime (listed on page 11-13) as well as a version of Rails. The version is very critical and will become more complex as the alternative Ruby runtimes reach maturity (JRuby, IronRuby, MacRuby, Rubinius, Maglev) and new versions of Rails appear.

The operating system (usually some version of [Linux](#)) and the Web server (usually [Apache](#)) also are version sensitive and in the case of the Web server there is usually a configuration process involved.

A compatible database is an absolute requirement (see page 28). Follow this with the need for some tool to actually deploy the application (usually [Capistrano](#)), and then you also need a piece of software called an *application server* (usually [Mongrel](#) or Passenger) to tie everything to make it all work together.

SEE THE PROBLEM?

Man-handling all of these parts and pieces has been a labor of love performed by developers in start-ups. But even in start-ups, once you get beyond a few Rails applications, it becomes a real task to keep up with everything deployed, where they are deployed, the versions against which they are deployed, etc.

Streamlining this entire process is an ongoing effort in the Ecosystem. Certain of the components innovate and improve independently. An example of this is [Phusion “Passenger”](#).

PLAYERS

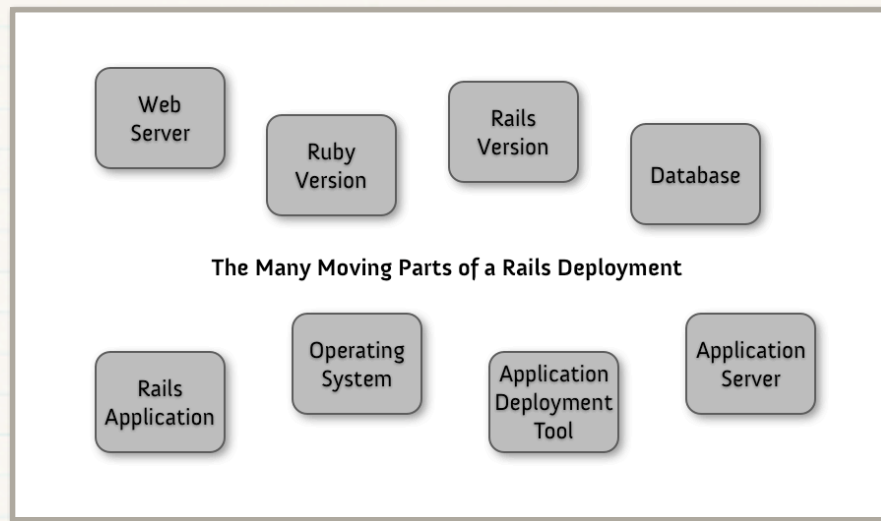
Application Hosting Providers
(Per Page 32)

Open Source Libraries

KEY POINTS

- ★ Deployment of created Rails applications requires lots of knowledge to synchronize all the components
- ★ Most of the innovation in this area has been at the component level
- ★ Current solutions rely on very savvy technical staffs, consultants, or hosted solutions providers
- ★ Much needs to be done to simplify the process for the mass market and address the needs of security, organization, and control

Implementation Tools: Deployment



HOSTED CLOUD-BASED APPROACHES

The first group within the Ecosystem that has addressed the deployment awkwardness of Rails has been those companies (usually venture funded) that offer application hosting services (see next page).

As part of their hosted offerings, they offer some streamlining of the deployment mechanics. It is not perfect and still usually requires a very savvy developer on staff to manage it as well.

These hosting services are so-called “public cloud” offerings in that they operate the server infrastructure (either their own or a third party’s like Amazon’s EC2 service).

These services have attractive pricing plans usually on some periodic basis (month, quarter, etc.).

CLOUD CHALLENGES

All of these public services however present a problem for many Enterprises, particularly publicly traded ones. Due to governance issues, they need extensive audit trails, security, and other compliance support. There is also the matter of control. Most need to manage things behind a firewall.

This has given rise to attempts at creating “private clouds” where the Enterprise either extends their own server infrastructure with “cloud” technologies or buys some “public” version offered by one of the hosting companies that provides some segmented and usually encrypted area for their data on those providers servers.

FURTHER STREAMLINING REQUIRED

Deployment remains an area where much work is to be done, and one of tremendous opportunity.

ACTIVITIES

Most of the institutional investment has been in hosting companies (next page) that offer some level of deployment assistance.

Groups within the Ecosystem continue to innovate on certain pieces of the deployment equation.

Piecing all the components together is still a manual process for most.

TRENDS

The need to simplify and streamline this process is critical to the growth of Rails within Enterprise clients, particularly those with IT staffs and organizational structures that share responsibility for the creation of new applications—either products or services that are Web-based.

OPPORTUNITIES

✓ This is perhaps the most fertile investment opportunity segment in the Ecosystem.

✓ The commercialization of deployment solutions will “harden” Ruby & Rails applications to make them more appealing to Enterprise clients.

Implementation Tools: **Hosting** (Applications)



Photo © James Duncan Davidson

...

SEGMENT BACKGROUND & HISTORY

Rails applications need to run on a server somewhere. Once the applications get to a certain size or function, the current popular means of running them is on a hosting provider that runs a data center or server farm that sells access to their machines. Storage and support services are customized in various ways, in this case to Rails.

Corporate or Enterprise users generally have access to in-house server facilities in order to maintain data in the applications behind firewalls. Start-ups do not have this infrastructure typically, which is why they prefer outsourced hosting providers.

For that reason most of the players listed cater to start-ups, consultants and independent developers or groups within enterprises that operate independent of a corporate IT department.

HOSTING PLUS SERVICES

Most all the players in this segment provide some differentiating features or services or target vastly different segments of the market. In some cases they offer application performance management and monitoring services (see page 29) with their own products or others. Most offer services traditionally associated with hosting providers such as configuration, testing, debugging, optimization, production server management, etc.

Some vendors run their own data centers. Others use outsourced ones including Amazon's Web Services infrastructure. Pricing is based on features, services, control functions, and storage.

PLAYERS

Blue Box Group

Engine Yard

Heroku

Joyent

Rackspace

Rails Machine

Rails Playground

RightScale

KEY POINTS

- ★ Hosting providers lead the Ecosystem in outside investment
- ★ Enterprise clients are assessing the viability of "cloud computing" though security issues, control, and uptime remain challenges
- ★ CIOs are hopeful such hosting services will eventually allow them to cut software licensing fees and infrastructure costs

ACTIVITIES

This segment has received the vast majority of outside investment in the Ruby and Rails Ecosystem due to the hardware and facility requirements. This allows several of these companies to be the principal sponsors of conferences, which helps grow and stabilize the community.

A few of these companies have the highest profiles of any in the Ecosystem thanks to their venture investments.

TRENDS

This segment is perhaps the most competitive in the Ecosystem.

The offerings vary widely though all follow a "cloud" computing approach where the applications are hosted at the facilities of the listed companies' server farms (or in some cases, outsourced facilities). This creates issues for many enterprise companies that need to keep critical data behind their firewalls.

OPPORTUNITIES

✓ It is too early to tell which companies will dominate as the Ecosystem grows. Further investment will likely come as follow-on rounds.

✓ Cloud computing is an investment darling at the moment, but Enterprises struggle with the concept because of control and security issues that will eventually limit large-scale growth until solutions are found.

Implementation Tools: **Hosting** (Applications)

| COMMERCIAL COMPANIES | |
|--|--|
| Blue Box Group http://www.blueboxgrp.com | <p>A Tacoma, Washington-based private company founded in 2003. It later moved to become a full-service provider for hosting Rails applications. Determined to stay small and profitable, it is seeking to avoid venture investment. Recently introduced an API solution with the goal of combining the “flexibility of cloud computing with the power of traditional hosting.”</p> |
| Engine Yard http://www.engineyard.com | <p>The most highly visible start-up on the list with \$37.5 million in venture investment (two rounds in 2008, one in 2009), this San Francisco-based company also employs several of the luminaries in the Ruby/Rails Ecosystem and contributes to the Open-Source Rubinius Ruby implementation (see page 13). The company is a major sponsor of conferences in the Ecosystem. Services are charged by the “slice” for managed Rails hosting and deployment.</p> |
| Heroku http://heroku.com | <p>A two-year-old Y Combinator start-up based in San Francisco, California, which raised \$3 million in venture funds in 2008 from Redpoint Ventures and Angels. It has a slightly different angle by offering an online Rails deployment system in combination with hosting (which is provided through Amazon Web Services). Estimates are that 25,000 unique Ruby applications have been deployed to date on the platform.</p> |
| Joyent http://www.joyent.com | <p>Targets a broader audience with PHP, Python, and Java application hosting in addition to Rails, which is reflected in the second largest number of hosting clients on this list (4,000). In 2005 it acquired TextDrive to help power its hosting services and partnered with Dell to provide over \$3 million worth of free hosting services for Facebook developers. Based in Sausalito, California, and self-funded since 2004.</p> |
| Rackspace http://www.rackspace.com Slicehost http://www.slicehost.com | <p>The largest company on the list with 2,500 employees, it is the only publicly traded one on this list. Rackspace is global and is known for their high-end managed hosting and dedicated services. Headquartered in San Antonio, Texas, with eight worldwide data centers. It hosts 33,000 clients (compared to an average of < 500 for others on this list). Slicehost, a Rackspace company, has become quite popular in recent months with developers by offering “slices” of servers.</p> |
| Rails Machine http://railsmachine.com | <p>Founded in 2006 and run by a small, focused team based in Savannah, Georgia. Emphasis is on creating the ultimate hosting environment for the Rails community, particularly targeting developers building commercial Rails applications. Offers a range of services.</p> |
| Rails Playground http://railsplayground.com | <p>A hosting provider geared toward developers. Established in 2005, it was one of the first Rails-specific hosting companies. It is a service owned and operated by Hosting Playground, Inc., a privately-owned Myrtle Beach, South Carolina, corporation.</p> |
| RightScale http://www.rightscale.com | <p>A Santa Barbara, California, company that raised \$17.5 million of venture funding in two rounds in 2008 (second largest funding in this category for a start-up). Supports a number of cloud-computing infrastructures including Amazon EC2, GoGrid, Rackspace, and others to allow clients to avoid “lock-in” to any cloud vendor with its Cloud Management Platform.</p> |

Implementation Tools: Hosting (Code)



...

SEGMENT BACKGROUND & HISTORY

In contrast to the commercial hosting of applications, most Open Source software projects, including Ruby and Rails, are available for free to download from a number of sites whose sole purpose is to act as a free forum for developers to list, manage, and make available their coding projects.

These sites are funded and supported by various organizations, or through premium paid plans, and have been an essential function in the development of the Ruby and Rails Ecosystem.

MAIN PLAYERS

The two players in this segment that are the main focus in the Ecosystem are RubyForge, the original source of Ruby and Rails, and the hot newcomer GitHub, which has established itself as

a social network of developers of sorts. Each offers compatible and complementary features and some important differences that dictate developer preferences.

VALUE & USE

Developers create Open Source projects and post them to these sites for download by the broader community. This means valuable libraries of code that perform certain functions or solve distinct problems are made available under various Open Source licenses so that they can be freely used (usually) by other developers in their projects.

Such projects are often maintained on these sites by teams of developers authorized by the original project creators to contribute changes.

The numbers of projects are impressive, numbering in the thousands. Registered users also number in the tens of thousands on Ruby-specific sites like RubyForge to millions on the general-purpose (all

PLAYERS

Bitbucket

GitHub

RubyForge

SourceForge

KEY POINTS

- ★ Code hosting repository sites are fundamental to the functioning of the Ecosystem and are THE source of Ruby implementations, Rails and associated libraries, frameworks, and projects
- ★ The vast majority of the projects on these sites are Open Source carrying no licensing fees, no proprietary lock-in, and lower operating costs
- ★ Open Source software projects have achieved equal or greater reliability and stability to proprietary commercial software due to the broad international community freely supporting and contributing to them

language) sites like SourceForge.

ACTIVITIES

Open Source code hosting project sites have been a staple of the Open Source movement and the principal source for downloading projects, usually without any sort of licensing fees.

The Ruby/Rails Ecosystem started with an exclusive Ruby/Rails site (RubyForge) in 2001. Projects are now also hosted on a number of such sites that do not focus exclusively on just one language.

TRENDS

Code hosting sites continue to grow in size and functionality.

Most such sites are either funded by an external source supplemented with Open Source contributions and/or through paid premium packages, usually for private code repositories.

A new player, GitHub, has introduced social networking functionality.

OPPORTUNITIES

✓ This is not really a segment offering profitable potential investments.

✓ At best these sites will do well to cover their operating costs while offering an invaluable service.

✓ Providing links and connectivity to the services on several of these sites will serve to streamline their utility with Enterprise developers.

Implementation Tools: Hosting (Code)

| NON-PROFIT & COMMERCIAL ENTITIES | |
|---|---|
| Bitbucket http://bitbucket.org | A minor player for the Ruby/Rails Ecosystem. It has 10,000+ users and is primarily a source site for Python and Java code hosting projects. It hosts its own version control system, Mercurial, which is a <u>distributed</u> version control system that allows loosely coupled work groups. This site does offer paid plans for private repositories. |
| GitHub http://github.com | The hot new entry in this segment, its hosting service is for projects that use the Git revision control system. Written in Rails by Logical Awesome developers Chris Wanstrath, PJ Hyett, and Tom Presont-Werner, the site provides social networking functionality to display how developers work on their versions of a repository. Its version control system, based on Git, is also a <u>distributed</u> one that is used by loosely coupled workgroups. It has now become the principal source for Rails where Rails versions are maintained by its core team on the site. The site has 46,000+ projects and 90,000+ developers. Ruby/Rails projects represent a minority of the projects as it has become the Open Source code hosting site for most languages and frameworks. The unique social networking functionality is the primary driver for its popularity. It also offers paid plans for private repositories where public ones are free. |
| RubyForge http://rubyforge.org | The original source site for the Ruby & Rails community created and funded by the non-profit Ruby Central, Inc., which was formed in 2001 to promote the Ecosystem through this site and the major annual Ruby and Rails conferences. RubyForge focuses exclusively on the Ruby & Rails community unlike the others on this list that host projects from other languages and frameworks. There are nearly 8,000 hosted code projects on the site and 37,000+ registered users. It is the primary download site for Ruby implementations. It supports Subversion, which is an Open Source <u>centralized</u> version control system particularly critical to Enterprise development. The site also supports Git and CVS version control systems. (RubyForge is operated by InfoEther, the author of this paper, under an arrangement with Ruby Central, Inc.) |
| SourceForge http://sourceforge.net | The largest Open Source software development site with over 230,000 registered software projects. It covers all languages and frameworks. Registered users stand at 2 million. SourceForge.net is owned and operated by SourceForge, Inc., a US publicly traded company that runs the community generated content site Slashdot and the online tech bazaar ThinkGeek. For the Ruby & Rails Ecosystem, SourceForge is used primarily as a duplicative or backup site. |

Technology Tools

BACKGROUND

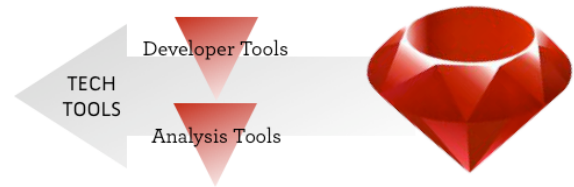
This is a branch of the Ecosystem that evokes strong passions in the developer community.

The products and Open Source projects listed are the shop tools with which the technology is built. Developers form strong attachments to various ones and can get into deep religious arguments over their use and relative merits.

SOME REPURPOSING INVOLVED

A number of the tools in this segment have been brought forth and adapted from other programming languages by their vendors. The modifications sometimes leave much to be desired. Others are making strides that improve older products, and some truly do stand out.

There actually has not been significant external investment in this area particularly when compared to the investments in companies and their products that focus on the implementation tools for the Ecosystem.



Several of the companies involved only address the Ruby and Rails marketplace as an adjunct to the bigger and older programming language marketplaces their product lines normally address, mostly being Java-related.

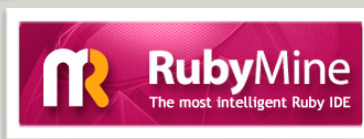
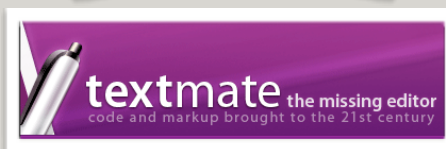
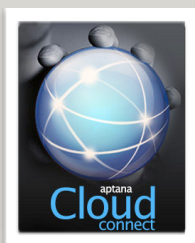
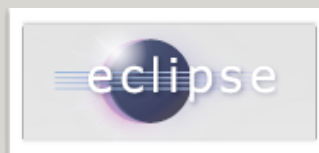
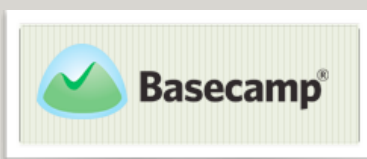
NOT ALL CODE-CREATION RELATED

Some tools, such as the Agile development products like Basecamp, Mingle and Rally are also mostly language agnostic. They are designed to facilitate the development process of the business problem being addressed and therefore have a broader market appeal.

OPPORTUNITIES?

The tools used for code analysis, however, have no companies backing their creation. They are exclusively today pure Open Source contributions by dedicated individuals.

Sample Product Logos



Tech Tools: Developer Tools



...

SEGMENT BACKGROUND & HISTORY

Software developers use a number of tools to create, test, debug, and run code. This section will group “text editors,” “Integrated Development Environments (IDEs)” and “project management systems” together if they all share some common functions used in software development. In many cases these tools are used in conjunction with each other.

In the Ruby and Rails Ecosystem the most popular developer tool currently is TextMate (by MacroMates), which is a text editor for the Mac OS X operating system. This is because the majority of developers in the community use Apple Mac computers.

There are a number of IDEs available. Some are Open Source, and others are

commercial products. An IDE usually consists of a source code editor, a compiler and/or interpreter, build automation tools, and a debugger. The hottest new entry in this area is RubyMine by JetBrains.

A new integrated Open Source project management tool, Redmine, was recently introduced that is a combination of project management and bug tracking, and it is being used increasingly in conjunction with text editors and/or IDEs.

PASSION & LOYALTY

Since these tools are essential to the day-to-day activity of software developers, passions run high as to the best or most useful. Many really experienced Ruby coders actually disdain the use of the older full-on IDEs, which they see as being overdone and re-purposed from other languages, but new entrants are trying to turn that perception around.

PLAYERS

Open Source

Active Reload/ENTP

Aptana

Embarcadero

JetBrains

MacroMates

Rally Software

SapphireSteel

ThoughtWorks

37signals

KEY POINTS

- ★ Developer tools consist of many components and are the essential tools of programmers
- ★ Other languages have developed established Integrated Development Environments (IDEs), but they have not yet achieved similar popularity in the Ruby & Rails community as non-IDE tools are being used in combination in the Ecosystem by most experienced developers

ACTIVITIES

Development in Ruby and Rails is being done with a combination of cost-effective programming tools that depart from the more structured traditional IDEs being adapted from other languages.

IDEs tend to be used by developers new to the community who come from more traditional backgrounds or are more comfortable working in a familiar tool.

TRENDS

Development environments originally created for other languages have been adapted to Ruby and Rails with varying degrees of acceptance.

The majority of experienced developers are using tools that are Open Source or have modest commercial costs.

New tools and improvements are being continually introduced.

OPPORTUNITIES

✓ Though established players are fairly entrenched in terms of developer loyalty, there are opportunities for improvements to existing offerings and the introduction of new, more integrated tools.

✓ Combining one or more of these tools into integrated packages with deployment tools is a fertile opportunity as the Ecosystem expands.

Tech Tools: **Developer Tools**

OPEN SOURCE TOOLS: INTEGRATED DEVELOPMENT ENVIRONMENTS (IDES)

| | |
|---|---|
| NetBeans (Open Source—Sun) http://www.netbeans.org | One of the original Java IDEs released as an Open Source product and run by a non-profit organization set up and sponsored by Sun Microsystems. It supports a broad range of languages and other technologies. |
| Eclipse (Open Source—Foundation) http://www.eclipse.org | An older IDE released to Open Source and maintained by a consortium of companies including Borland, IBM, MERANT, Red Hat, Rational Software, and others. It is a well-known IDE in the Java world with a large ecosystem built around it. |
| Aptana RadRails (Open Source—Aptana) http://www.aptana.com/rails | A long-established free IDE plug-in to Eclipse optimized to Ruby and Rails is offered by Aptana, a California company specializing in developer tools for several languages. There is also a “cloud” service version offered as a commercial project (see next page). |

OPEN SOURCE TOOLS: PROJECT MANAGERS

| | |
|--|---|
| Redmine (Open Source—Independent) http://www.redmine.org | A relatively new flexible project management Web application actually written using Rails. It is cross-platform and cross-database. It was created and is maintained as an Open Source project/organization by Jean-Philippe Lang. In the period of less than a year it has been available, it has achieved considerable buzz in the community. |
|--|---|

COMMERCIAL PRODUCTS: TEXT EDITORS

| | |
|--|--|
| TextMate (MacroMates) http://macromates.com | Offered by MacroMates (Copenhagen, Denmark), TextMate is the virtual default tool used by a majority of developers in the Ruby & Rails Ecosystem. It is only for the Apple OS X operating system. It is usually used in conjunction with project management systems and other tools. |
|--|--|

COMMERCIAL PRODUCTS: PROJECT MANAGERS/ISSUE TRACKERS

| | |
|---|--|
| Basecamp (37signals) http://www.basecampHQ.com | The leading Web-based project collaboration tool by the Chicago-based 37signals. Rails grew as an Open Source project out of the Basecamp effort, and the creator of Rails is a partner in 37signals. Basecamp is used for more than just software projects and has millions of users for its Software-as-a-Service (SaaS) model. There are related packages to this service that are also available—Highrise, Backpack, Campfire. |
| Lighthouse (ActiveReload/ENTP) http://lighthouseapp.com | A popular Web-based SaaS product used for issue (i.e., bug) tracking that is available for a fee by ActiveReload, LLC, a small company founded by Rick Olson and Justin Palmer. The company is related to the consulting firm ENTP (see page 26). It has found a following and is used in conjunction with other tools. Creator Rick Olson is on the Rails Core Team. |

CONTINUED ON NEXT PAGE

Tech Tools: Developer Tools

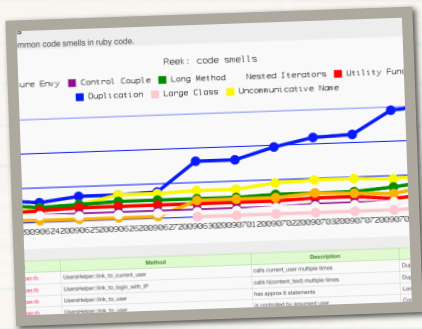
COMMERCIAL PRODUCTS: PROJECT MANAGERS/ISSUE TRACKERS (CONTINUED)

| | |
|---|--|
| <p>Mingle (ThoughtWorks) http://studios.thoughtworks.com/mingle-agile-project-management</p> | <p>Offered by a division of one of the largest Rails consultancies (see page 26), Mingle is an Agile development tool for project development and collaboration. It supports all the standard Agile approaches including workflows and “scrum” and uses customizable wiki and “card wall” techniques. It is now in its 2.0 version and has built up a sizable international community.</p> |
| <p>Rally Agile (Rally Software) http://www.rallydev.com</p> | <p>A well-established venture funded company based in Boulder, Colorado, Rally is a comprehensive SaaS service for online management of projects, programs and products utilizing Agile development methods. It is more involved than other products in this segment as it addresses a broader market. It is used on numerous Rails projects.</p> |

COMMERCIAL PRODUCTS: INTEGRATED DEVELOPMENT ENVIRONMENTS (IDES)

| | |
|--|---|
| <p>RubyMine (JetBrains) http://www.jetbrains.com/ruby/index.html</p> | <p>Built upon the proven IntelliJ platform (see next), the RubyMine IDE has achieved considerable buzz in the community when it was released from beta in April 2009. It bills itself as “the most intelligent Ruby IDE.” The company JetBrains is headquartered in the Czech Republic with offices in the US. It was originally founded in 2000.</p> |
| <p>IntelliJ IDEA (JetBrains) http://www.jetbrains.com/idea/index.html</p> | <p>An established IDE for the Java world that has been used for Ruby development. It was one of the early products that established JetBrains (see entry above) and has achieved a healthy following with Java programmers, many of whom are transitioning to Ruby and Rails projects.</p> |
| <p>3rd Rail (Embarcadero) http://www.embarcadero.com/products/3rdrail/index.php</p> | <p>A feature-rich, high-powered IDE written from the ground up for Ruby and Rails development by Embarcadero, a large company with many international offices that develops multi-platform developer and database solutions for a number of languages and technologies.</p> |
| <p>TurboRuby (Embarcadero) http://www.embarcadero.com/products/turboruby/</p> | <p>A lighter-weight IDE (in relation to 3rd Rail) also from Embarcadero. It was built from the ground up too in order to streamline Ruby development with high-efficiency coding and visualization tools.</p> |
| <p>Ruby In Steel (SapphireSteel) http://www.sapphiresteel.com</p> | <p>An established Ruby IDE by SapphireSteel Software, a division of Rosedown, Mill, Ltd., that develops the Steel IDE for programming Ruby and Flex in Microsoft’s Visual Studio. The company is working on a number of enhancements and new versions of the Steel IDE.</p> |
| <p>Aptana Cloud Connect (Aptana) http://www.apтана.com/cloud</p> | <p>A SaaS product available as a monthly service where Web applications can be developed, hosted, and managed. It uses or is compatible with the Eclipse IDE plug-in by Aptana (listed on the previous page).</p> |

Tech Tools: Analysis Tools



...

SEGMENT BACKGROUND & HISTORY

This segment is primarily used for generating metrics reports for the analysis of Ruby code.

A Ruby Gem (a packaging approach for Ruby code) designated "metric_fu" located on the Ruby project site, RubyForge, aggregates the major ones used by programmers. It is also available on the GitHub repository.

SUMMARY

All of these tools can be reviewed at <http://metric-fu.rubyforge.org/> or http://github.com/jsruggs/metric_fu and some of the main ones are also summarized as follows:

Coverage: A report that runs all the tests in a test folder and specifications in a specifications folder.

Saikuro: A Ruby program to calculate the "cyclomatic complexity" of Ruby source code. That means it determines the number of linearly independent paths through a program's source code.

Floy: Analyzes Ruby code for structural similarities.

Flog: Another way of measuring complexity (or "tortured code") as the authors put it.

Reek: Detects common "code smells," that is, any symptom in the source code of a program that possibly indicates a deeper problem.

Roodi: Warns of design issues in Ruby code.

Stats: Gives statistics about an application at different points in time.

Churn: Finds files that change a lot, which is sometimes an indication of deeper problems.

PLAYERS

No Commercial Companies

Numerous Open Source Projects Hosted on RubyForge.org & GitHub

KEY POINTS

- ★ Analysis tools are used by programmers to find problems or potential problems in their code
- ★ The tools that exist in this category are Open Source & free to use
- ★ Such tools are created & provided by dedicated members of the developer Ecosystem
- ★ In most cases these tools are rolled into other processes or products

ACTIVITIES

This is an area of modest activity and is motivated by developer community commitment.

The growth of more tools in this segment will continue as the need arises and dedicated developers undertake their creation.

TRENDS

Tools in this segment continue to evolve and are updated to support changes to the main versions of Ruby and Rails.

Commercial and investor-funded companies have not seen this an area of significant interest, though many of these existing tools are used in their code development activities.

OPPORTUNITIES

✓ This segment remains relatively static.

✓ There are opportunities to develop easier to use tools in this category or to incorporate existing tools into easier developer tool interfaces for programmers new to the language.

Recap

...

OVERVIEW

The goal of this paper was to educate and serve as a source document. Hopefully we have been able to provide sufficient explanation and have kept the discussion understandable for a non-technical audience.

Ruby was created to be a programmer's language. That fact is critical to the future because the passion and affection developers have for it is its most powerful asset. That created the Ecosystem, and that is what will likely allow the language to spread in popularity over the several other competing newer languages and the older established ones.

Rails was created in order to solve a real business problem and grew into a major development tool that was promoted and embraced by Web 2.0 start-ups and Agile-minded businesses alike. It spawned a host of copycat clones in other languages, none of which had the same capabilities or impact. With the inclusion of the Merb framework in what will be Rails 3.0, it will evolve into one of the most powerful tools available for creating value in applications written for the Web—internal to an Enterprise and external as products or services to the commercial or non-profit worlds.

ENTERPRISE ACTIVITY

At the most recent Rails Conference in Las Vegas, the audience of 1,300 software developers was asked by a show of hands:

- * How many were doing Ruby/Rails work in start-ups?
- * How many were doing Ruby/Rails work as independent consultants?
- * How many were doing Ruby/Rails work in Enterprises?

A third raised their hands in each category. This was a milestone, because in the past several conferences, only a few were identified as working in Enterprises. Considering that a third were also working as consultants and guessing that a substantial portion of their clients were Enterprises, one can make the assumption that fully half of all Ruby and Rails activity is now being done in Enterprises, which opens up a whole new market.

In its required annual filings in 2009 with the Securities & Exchange Commission (SEC), Microsoft listed both the Ruby language and the Rails framework as significant competitors.



KEY TAKE-AWAYS

Surge in Open Source

Interest: The economic downturn has focused increased interest in Open Source technologies to reduce costs, escape proprietary applications, and spur innovation—benefiting the Ecosystem.

Critical Juncture: Like languages before it, Ruby (and Rails by extension) has entered a phase that portends significant growth—moving from the early pioneers to the mainstream. Think of it as being equivalent to where Java was in 1999 before its dramatic growth.

Developer Preferences: Ruby and Rails are easier to use and less arcane in syntax with some very powerful, efficient features—programmers do not have to fight them to use them, which generates a degree of affection that translates to staying power.

Opportunities: A great deal of effort and investment has gone into stabilizing and commercializing both Ruby and Rails, but much more needs to be done and opportunities abound.

Key Values: Ruby and Rails are suited to Agile processes where applications can be built quickly and efficiently to therefore realize greater value faster. Supporting, maintaining, and evolving these applications are also less costly.

MAJOR OPPORTUNITY LIST

✓ **Training:** More developers moving to the language and more Enterprise interest will generate a greater need for comprehensive training courses directed at programmers *and* management.

✓ **Consulting:** The ones that will expand from the current crop are those that can adapt to the organizational challenge of Enterprises.

✓ **Deployment:** Still the trickiest area that needs to be solved in better ways—inside and outside the firewall.

✓ **Cloud Computing:** The recipient of the majority of outside investment, it still needs to figure out how to deal with the “public” vs. “private” cloud issues that will determine future success in the Enterprise.

✓ **Runtime Customization:** Ruby runtimes (JRuby, IronRuby, MacRuby, etc.) are reaching maturity. Building custom tools for each will offer big rewards.

“Ruby on Rails is astounding. Using it is like watching a kung-fu movie, where a dozen bad-ass frameworks prepare to beat up the little newcomer only to be handed their asses in a variety of imaginative ways.”

~ Nat Torkington, Open Source Conference Chair

COMMERCIAL “HARDENING” PROCESS

The next major growth phase for the Ruby and Rails Ecosystem involves packaging, simplifying, and consolidating various parts and pieces into more “hardened” commercial products and services.

The wider market will not have business people or developers who want to struggle with all the various Open Source components to make a Rails application run. They do not want to worry about compatible versions, and they want support. Enterprises want to realize results with speed and without undue hassles.

Up until now, many of the hassles were actually enjoyed by the troops working in start-ups. A start-up programmer loved to code up the application, select the proper runtime version, configure the servers, and man-handle the appropriate deployment library.

Enterprises are organizationally more complex. A business unit may champion an application and can even code it up, but separate Q&A testing groups need to be involved, and the final version needs to be handed off to an IT staff to run, maintain, and support it.

Packaging and simplification is thus the key to this looming market segment.

A POWERFUL ECOSYSTEM IS KEY

The Ruby and Rails Ecosystem that has been built up over the last eight years is now mature enough to address this market expansion. It will only continue to grow and become more savvy.

This process is already in motion. More investment in many areas in the Ecosystem will serve to accelerate things and bring more innovation to the market faster.

BUT WILL IT WIN?

Investors always want to bet on the winning horse. Many are aware of the impact and potential of the various new programming languages and their frameworks—Ruby & Rails, Python & Django, PHP, etc.—and the continued use of Java and its mature ecosystem.

There is not enough classic marketing data to confirm who is ahead. In fact, several may grow and co-exist, and no new player will truly dominant.

However, one thing is certain: the new ones are better, faster, and less expensive—and that is why the older languages will give way eventually.