Evaluating web development frameworks:
Django, Ruby on Rails and CakePHP

Julia Plekhanova
Temple University

Fox School of Business
TEMPLE UNIVERSITY ®

The IBIT Report

Foreword

Welcome to the new and improved "The IBIT Report." A key goal of the Institute for Business and Information Technology is to produce cutting edge knowledge to create and sustain excellence in information technology. To meet that goal, The IBIT Report will produce periodic "white papers" that are based on rigorous neutral research and are written to provide actionable knowledge for information technology practitioners. Each report will focus on an important topic that is of interest and relevant to the members of IBIT, is based on strong academic research, and provides added value from an academic perspective. The IBIT Report was started in 2008 with the release of reports on social computing, wireless technologies, and politics on the Internet.

I am pleased to announce a new chapter in the evolution of The IBIT Report. The report now has a formal editorial structure and process. The editors and editorial board will consult with IBIT members, industry representatives, and academic professionals to identify potential topics and authors. The editorial board will also actively participate in assessment and review of each report prior to final publication.

Evaluating Web Development Frameworks is the first report that is being published under this new structure. Senior information technology managers and experienced developers have a myriad of options available to them in developing web applications including building from scratch to using a commercially available content management system. But there is another option: using an open source web development framework. Evaluating Web Development Frameworks looks specifically at three leading open source web development frameworks and provides a methodology to evaluate each framework based on your own unique context. We believe the report provides an insightful analysis of this option and a tool that can be applied in your environment.

Bruce Fadem
Editor-in-Chief
September 10, 2009

Introduction

Today, websites are complex applications that perform transactions, present real-time data, and provide interactive user experiences. Web based software is becoming as powerful and as important as desktop software. Developing web applications that provide advanced functionality is now a complex task that involves multiple developers, evolving toolsets, and many options. Web frameworks provide a golden mean between building an application from scratch and using an out-of-the-box content management system (CMS). Using an out-of-the box CMS is like buying a finished house. You can choose the paint and furniture, but ultimately you have to trust that the house was made to suit your needs infinitely. Building from scratch is like going to the store and buying everything starting from nails and finishing with plumbing and wooden panels. Using a web framework lets you choose between different types of walls, different types of wiring, plumbing, and windows that can be pieced together in a way that best suit your needs [8].

Open source continues to gain popularity. An open source project often develops rapidly because a large number of developers from around the world contribute to it. Since the code is visible to the whole community it is often very clean documented code. Open source tools also avoid vendor lock-in. Companies choosing open source not only own the source code but all the data. Further, open source tools typically have a large support community who can rapidly react to problems and provide assistance.

This report focuses on three leading open source web development frameworks: Django, Ruby on Rails and CakePHP written in three different languages – Python, Ruby and PHP respectively. All three frameworks have similar architectures and claim to have similar characteristics, such as greatly enhanced productivity and code re-use.

This report is designed for senior managers and experienced developers considering the development of complex web applications and interested in assessing the feasibility of open source web development frameworks. The report provides a methodology to evaluate each framework. The methodology, criteria, and weights provided in this report are generic and comprehensive. Each organization should adapt the methodology of this report to its own unique context.

Architecture

All three web application frameworks share the following features and attributes:

- Streamline development process by automating some of the parts;
- Add structure to the code and make code more readable;
- Reuse components, to speed up the development process;
- Support concurrent creation/update of content and development.

The three frameworks share the model-view-controller (MVC) architecture. Although Django calls its architecture model-view-template (MVT), it is very similar to MVC. In MVC, an application's data model, user interface, and control logic are separated into three components. The model manages the data of the application and the business rules; the view is responsible for displaying data to the user through an interface; and the controller interprets user inputs and communicates with the model to make the appropriate changes [14]. Exhibit 1 illustrates MVC and the interaction between the components.
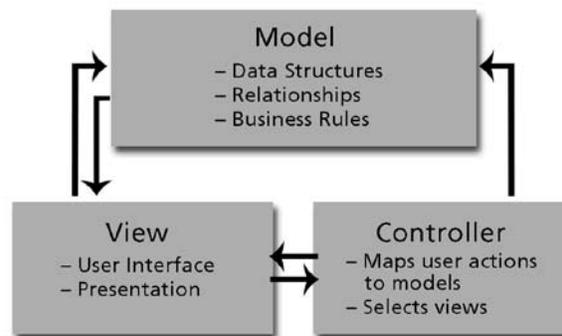
Exhibit 1 illustrates MVC and the interaction between the components

Rails, Django and CakePHP frameworks are relatively new, open source toolsets. All emerged from real world websites, are proven to handle very large volumes, and encourage agile development (quick, incremental updates). All three frameworks currently support commercial high volume sites. Both Rails and Django have implementations that also work with .Net and Java - see JRuby, IronRuby, Jython and IronPython for more information. All three frameworks work with a wide range of databases and web servers.

Overview of Django, Ruby on Rails, and CakePHP

Django

Django is a full-stack Python "web framework that encourages rapid development and clean, pragmatic design" [7]. Developed in a fast-paced online-news environment in Lawrence, Kansas, it claims to be "the web framework for perfectionists with deadlines." Django was developed in 2003, but wasn't released as open source until July 2005, the formal release 1.0 was in September 2008, and the current version is 1.1 (June 2009) [7]. "Django focuses on automating as much as possible and adhering to the DRY (don't repeat yourself) principle" [7]. Djangosites.org serves as a showcase of Django powered sites; there are 2383 websites listed, including news and entertainment sites such as Lawrence (http://lawrence.com), Washington Post (http:// washingtonpost.com), LJWorld.com, (http:// LJWorld.com), social networking and photo sharing sites such as Tabblo (http://tabblo.com), Grono (http://grono.net), as well as business, educational, ecommerce, and other sites. There are several CMS built in Django, such as Ellington, a CMS specifically built for newspapers, magazines, and entertainment websites, Django-CMS, and Django Page CMS. More detailed information, documentation, tutorials and downloads can be found on Django's official site, http://www.djangoproject.com.

Django is based on Python, which is a very popular programming language. Often compared to Perl, Ruby, and Java, it is a dynamic object-oriented language. Python offers strong support for integration with other languages and tools and comes with extensive standard libraries [17]. Large firms such as NASA, Google, YouTube, Yahoo!, and Apple use Python for their applications. The language is easy to learn and follows the principles of simplicity, readability, expressiveness, and modularity. Python is an open source language and has a large number of libraries and modules available. Django has a lot in common with Python. Python is overseen by the Python Software Foundation (PSF). Similarly, Django has started its own organization called the Django Software Foundation (DSF) [13]. In December 2008 a new version (3.0) of Python was released. In version 3 Python underwent a thorough clean up that gives it the opportunity to grow and develop going forward. This version is not backward compatible, and consequently it will take a long time until large projects transition to the new version [17].

Ruby on Rails

Rails was developed by 37Signals Inc., in Chicago, Illinois for a collaborative project management application called Basecamp. Rails was released as open source in July, 2004 and version 1.0 came out in December, 2005. The current version as of June 2009 is 2.3. Similar to Django, "Rails is a full-stack framework for developing database-backed web applications" [18] that is written in Ruby and follows the principles of agile web development, increases productivity and speeds up development. Basecamp (http://basecamp.com), Big Cartel (http://bigcartel.com), Shopify (http://www.shopify.com), 43 Things (http://www.43things.com), and others are all built on Rails. One of the highest-traffic Rails websites is yellowpages.com, serving approximately 100-170 million page views per month [20]. Tutorials, documentation, and downloads can be found on the official Ruby on Rails website http://rubyonrails.org.

Ruby is a "dynamic, open source programming language with a focus on simplicity and productivity. It has an elegant syntax that is natural to read and easy to write" [19]. Ruby is used by NASA Langley Research Center, Motorola, Lucent and other major firms. Ruby got its popularity from the Rails framework.

CakePHP

"CakePHP is a rapid development framework for PHP that provides an extensible architecture for developing, maintaining, and deploying applications"[5]. CakePHP was developed in 2005, when Ruby on Rails was gaining popularity. The goal of CakePHP is to enable users to rapidly develop robust and well-structured web applications. The current stable release is 1.2.3. Examples of websites "baked" with CakePHP include Firefox Add-ons (http://addons.mozilla.org), Scratch (http://scratch.mit.edu), Yale Daily News (http://www.yaledailynews.com), and The Onion Store (http://store.theonion.com). Documentation and latest news and releases can be found on the official CakePHP website, http://www.cakephp.org.

CakePHP is based on PHP, a general-purpose scripting language that is especially suited for web

programming. There is extensive support for this language and hundreds of web application examples. The language is easy to learn and understand, and many programmers write in PHP. The main principles behind PHP are robustness and simplicity. There are over 30 web frameworks that use PHP. Wikipedia, Flickr, Facebook, Digg, WordPress, PBWiki.com, and Friendster are all built in PHP.

Evaluation method

To evaluate a web development framework many factors should be considered. A web developer, project manager, or entrepreneur, may all have different perspectives and the specific needs will vary from one project to the next. Further, web development frameworks are influenced by a large number of constantly evolving factors, including social, perceptual, and contextual forces that do not neatly fit into a technical evaluation. These social factors are as important as technical factors especially for open source development which is highly dependent on "developer buy-in" and community support. It is therefore hard to assess each framework completely objectively.

Traditional metrics such as benchmark performance and the technical quality of the programming language and others are ignored as the differences today on these dimensions are increasingly minor and are less relevant for web development. Further, there are already evaluation metrics available using traditional criteria. To elicit the most useful set of evaluation metrics a blog was built with each framework. Blogging thus became a de-facto reference task. Blogs include the most essential features of almost any web application including authentication, content management, publishing, and maintenance. Building the blog prototypes helped provide hands on experience

> "Web development frameworks are influenced by a large number of constantly evolving factors, including social, perceptual, and contextual forces that do not neatly fit into a technical evaluation. "

with each framework and included the normal tasks faced by most web developers, such as installation, getting familiar with documentation, community support, customization, and extending features.

Based on the above analysis and hands-on experience, a set of seven evaluation items was developed. These items are evaluated below on a scale of 1.00 (Poor) to 5.00 (Excellent).

Evaluation

1. User interface development

This measure focuses on support for developing user interfaces. Specifically, each framework's default templates and support for JavaScript libraries were assessed. Templates are an important part of the MVC framework structure; they by default define the user interface of any new application built with the framework. JavaScript is a scripting language used in millions of websites and server applications and provides end-user interactivity.

Usability of templates (views)

Templates (in Django) or Views (in Rails and CakePHP) allow the respective frameworks to separate content from presentation. This paradigm and implementation of the MVC architecture is integral to each framework. Templates define the overall layout of the pages while individual page templates define the specific content of a certain page. The key difference between the frameworks is in the way they embed dynamic content.

Rails and CakePHP views use Ruby code fragments, and may therefore contain some complex functionality. Django, in contrast, uses a template language that can be understood by web designers who know HTML but have very little knowledge of programming. This could be important in projects where graphic and web designers are working simultaneously with programmers on creating the site.

"One major difference is that Django includes a template language" [6] whereas Rails and CakePHP do not. Therefore, the Rails and CakePHP views are more complex. The advantage of this approach is that it helps to reuse even small pieces of code.

"One of the core reasons for Django's inclusion of a template language is to separate application and business logic from presentation. There are several advantages to this philosophy, most of which stem from

the resulting separation of designer and programmer. Consider:

1. Designers can't break the application. Since they never touch the application code, there is no worry that a mistake by the designer is going to bring your app to a crashing halt
2. Designers have complete control to do literally anything they want with the content — and they can do it without having to bother the programmer at all.
3. Programmers aren't futzing with an application's design and front-end code. In short, this means no more cleaning up after your programmer, who doesn't seem to know that the <b> and <i> tags are deprecated.
4. Programmers can add to and change the application code without affecting the display" [6].

Further, the Django template language is not limited to HTML and makes it possible to create files in CSS, JavaScript, XML and other formats. If more advanced functionality is required (i.e. to print data in a 5 column table), the function is written in a view and pulled by a template. Therefore, Django template language stays clean and simple.

Django's approach thus offers more advantages for large projects, in which designers can easily understand the language and work independently on layout from programmers.

Support for JavaScript libraries

The three frameworks have different approaches towards JavaScript support. Although developers can use almost any JavaScript library, CakePHP and Rails have standardized on specific default toolkits - Prototype and Scrip.aculo.us - and include a number of helper functions. Django, on the other hand, leaves to the developer the choice of a JavaScript library. "Django includes only a JSON module, leaving JavaScript code and the choice of a JavaScript library to the developer" [2]. For developers who have an extensive knowledge of JavaScript, the Django approach seems to be superior. However, developers with less JavaScript experience will likely prefer the Rails and CakePHP approach. *(See Table 1.)*

	Django	**Ruby on Rails**	**CakePHP**
Usability of templates (Views)	Simple syntax; include template language	More complex syntax; include snippets of code	Complex; include snippets of code
Support for JavaScript libraries	Requires JavaScript, more freedom	Easy to use, less freedom	Easy to use, less freedom
Rating	4	3	3

Table 1: User interface development

2. Maintainability

This measure focuses on support for maintaining a site powered by the framework. In a multi-developer system, it defines how well changes can be tracked and restored; whether new development, updates, or installations create a problem for the existing pages. Websites are highly dynamic applications. Whether it is a blog or a company's website there is a constant need to update information. Google site ratings for example, rely on how often a site is updated. Therefore, in today's environment users expect that they will be able to update content and modify a site through a friendly interface.

Content management

All three frameworks provide a site administration interface that is used to perform CRUD (create, retrieve, update, delete) operations with data. Ruby on Rails and CakePHP offer just a basic functionality with scaffolding that can be further customized by a developer. Django, however, offers a more comprehensive solution. Django has automatically generated administrative pages, which can save time when developing an application. A developer has to add a few lines of code to get a well functioning back end. Exhibits 2 and 3 show the Django and Rails default administrative interfaces.

In Django, a developer passes attributes and parameters to the Django model classes and their fields. The generated pages have a range of options necessary for editing, to create, retrieve, update, and delete (CRUD) items in the model. Records can be also searched, filtered, and sorted. There is an option to look at the history page that lists all changes made to an object via the Django admin, with the timestamp and username of the person who made changes. In the environment where products have to go on the market fast, it means that once the administrative panel is up and running, a user can start adding the content through the back-end, while the developer can further work on setting up the views.

Automatically generated admin pages are helpful and save time, however they support only simple relationships. For advanced functionality, developers can create a custom administrative panel, overriding the default one. Therefore, this feature is more useful for smaller projects and websites than for large web applications.

Exhibits 2 and 3 show the Django and Rails default administrative interfaces.

Listing Posts

Title	Body	Author	Pub Date			
1st post	post boby	julia	2008-04-17	Show	Edit	Destroy
2nd post	post boby	julia	2008-04-17	Show	Edit	Destroy
3rd post	post boby	julia	2008-04-17	Show	Edit	Destroy

New Post

User authentication and session management

Each framework provides documentation for handling user accounts and permissions. Django Administrative panel includes handling of users and groups, and authentication. CakePHP, Django, and Rails all support session management which enables continuity between HTML pages.

	Django	Ruby on Rails	CakePHP
Content management	Automatically generated	Ok	Good
User authentication	Good	Good	Ok
User sessions	Supported	Supported	Supported
Rating	4	3	3

Table 2: Maintainability

3. Data management and migration

This measure focuses on how well history and backtracking are provided to reduce the risk of losing important data. This measure is essential to the development process. It defines how easy it is to roll back in case a problem occurs; it encourages a developer to experiment without fear of getting an error or losing data. This measure is also important when there is a team of developers working on an application.

CakePHP has a fairly good object relational mapping (ORM) framework. However it is not as advanced as the ones used by Django and Rails that are based on object oriented languages and therefore support more complicated cases such as inheritance. ORM for Rails is called ActiveRecord, ORM for Django and CakePHP is called Model.

Django has ability to create a database schema from the model definition in a model.py file. Currently, Django does not have a tool for database migration; however, Django developers are now working on this problem trying to find the best solution. The advantage of Django is easy integration with legacy databases.

Django ORM's ease of use, similar syntax to the rest of Django, and ability to handle 90% of SQL queries, makes getting started with Django easy. However, Django's ORM does not handle edge cases and complex queries well, and one has to write SQL by hand to make it work. SQLAlchemy handles many of these edge cases already, and it has been the Python standard for Object Relational Mapping. SQLAlchemy is the default ORM for most web based and non-web based projects. Whenever the need arises for an ORM, SQLAlchemy is used. Although SQLAlchemy can be used in place of Django's ORM, it is not the default choice, and still requires some

hacks to get integrated. In the early stages of Django development, SQLAlchemy was not an option, so a homebrewed ORM was necessary. Today Django developers are working on a better bridge between SQLAlchemy and Django. Python-based web frameworks, such as Turbogears and Pylons, already support similar behavior. In addition, there is a new breed of databases on the horizon, a non-relational database, including CouchDB, Google App Engine, BigTable, Cassandra, and many more. Being able to plug and play between multiple database mappers, both relational and non-relational alike is a big plus for any web framework.

Ruby on Rails has a powerful migration mechanism that allows creating and evolving model classes and the underlying schema without destroying the data. A migration script is a Ruby script that has access to methods that perform database operations. It also has access to model classes, so that the data in the database can be modified from within a migration as well. This script defines additions, modifications, and deletions to the database schema. "Optionally, the developer can also specify how data is to be migrated from the old version of the schema to the new version. Each migration script is assigned a version number, and changes to the schema can be rolled backwards and forwards by applying and

> **"Ruby on Rails has a powerful migration mechanism that allows creating and evolving model classes and the underlying schema without destroying the data"**

reversing the migration versions in sequence" [2]. Rails provides three databases – development, testing, and production. This approach is mature and supports complex development processes.

	Django	Ruby on Rails	CakePHP
Data management	Manual migration	Powerful	Weak, not safe
Object relational mapping	Yes, new features are under development	Yes	Yes, not as advanced
Rating	4	5	2

Table 3: Data management and migration

Rails has two main advantages in migrations compared to Django. Rails provides additional steps for deploying new changes for deploying new changes to already running applications without destroying data. Thus, if a database column's type is changed from integer to text, the Rails migration script would specify the steps required to move the data from the old integer column to the new text column. In Django, the developer would need to write an ad-hoc SQL script to perform this operation. The second advantage is the ability to easily roll back. Experimentation with models is possible in Django; however, since the data is not automatically preserved through such changes, it is more risky unless a developer uses a special mechanism for test data.

4. Testability

This measure defines the availability of debugging and testing tools as well as how easy and intuitive it is to solve development problems in each framework. Given that a significant aspect of web development is time spent on testing and debugging, it is important to assess how well a framework supports troubleshooting. This is especially important in the multi-platform, diverse, instantly global realm of web sites.

All three frameworks enable debugging in the development environment. When a web application encounters a problem, an error page is displayed. The page usually has a location of the error, the value of variables, a stack trace, and so on. Rails has a "debug" helper method. Both Rails and Django can work with integrated development environments (IDE).

CakePHP does not offer any tools that directly connect with any IDE or editor; however it provides several tools to assist in debugging and exposing what is running in an application. There is a function for basic debugging; however there is not too much information available on the function. Debugging was included in the 1.2 version manual for CakePHP, but it is not fully written yet and there are missing pages on error handling. Additional plug-ins such as Krumo can be helpful in debugging and troubleshooting.

Inside the code and templates a developer may dump the values of variables. Rails and Django provide methods for converting variables to strings and including them in the HTTP response buffer. A log file is another tool that is used by developers and all three frameworks have an ability of creating one to trace through problems.

CakePHP provides a test suite for version 1.2. Fixtures that are a part of the test suite are a way for a developer to test models by loading sample data. In order to run a test, a developer has to create fixtures and set up a test case. Django applications can be tested in a similar way by using fixtures.

Rails makes a full use of testing tools and includes a testing framework. Each time a programmer generates a new model and controller, Rails creates files in the project's test subdirectory that can be used for testing. Test scripts typically call the model and controller methods. All the tests in Rails are divided into unit tests for testing models, integration tests for testing the workflow, and functional tests for testing controllers. Similar to CakePHP and Django, Rails has test fixtures. However, Rails creates a separate testing database, whereas CakePHP creates special testing tables using the same database.

	Django	Ruby on Rails	CakePHP
Debugging	Good	Good	Ok
Testing tools	Some	Various	Few
Testing framework	No	Yes	No
Rating	3	4	2

Table 4: Testability

5. Popularity

This measure compares the underlying programming language that each framework uses. In many cases, a language is a personal choice of the web developer; however, some languages are more popular, more powerful, faster, and more widely used. Sometimes a developer feels very comfortable using a certain language and does not see a reason to learn another one. However, for a beginner developer it might be helpful to compare advantages and disadvantages of each language before deciding on which framework to use.

TIOBE, a company that is specialized in assessing and tracking the quality of software, created a list of 10 most popular programming languages. "The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. The popular search engines Google, MSN, Yahoo!, and YouTube are used to calculate the ratings". The index can be used to check whether one's programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system [21].

Figure 4: Popularity of programming languages

Position June 2009	Position June 2008	Delta in Position	Programming Language	Ratings June 2009	Delta June 2008	Status
1	1	=	Java	20.147%	-0.74%	A
2	2	=	C	16.779%	+1.27%	A
3	3	=	C++	10.594%	-0.21%	A
4	4	=	PHP	9.675%	-0.53%	A
5	5	=	(VIsual) Basic	7.943%	-1.84%	A
6	7	⬆	Python	4.756%	-0.14%	A
7	8	⬆	C#	4.536%	+0.48%	A
8	9	⬆	JavaScript	4.021%	+1.09%	A
9	6	⬇⬇⬇	Perl	3.909%	-1.64%	A
10	10	=	Ruby	2.629%	-0.01%	A
11	11	=	Delphi	2.182%	+0.16%	A
12	14	⬆⬆	PL/SQL	0.879%	+0.12%	A
13	26	⬆⬆⬆⬆⬆⬆⬆⬆⬆⬆	RPG (OS/400)	0.778%	+0.53%	A--
14	13	⬇	SAS	0.759%	-0.16%	A
15	15	=	Pascal	0.759%	+0.16%	A
16	27	⬆⬆⬆⬆⬆⬆⬆⬆⬆⬆	ABAP	0.726%	+0.49%	A--
17	12	⬇⬇⬇⬇⬇	D	0.620%	-0.83%	A--
18	16	⬇⬇	Lisp/Scheme	0.607%	+0.17%	A
19	19	=	Lua	0.557%	+0.19%	A
20	23	⬆⬆⬆	MATLAB	0.527%	+0.26%	A

From the chart we can see that PHP takes the 4ᵗʰ position, Python is on the 6th place, and Ruby is on the 10ᵗʰ position. These statistics have not changed significantly since June 2008.

"But nothing speaks to market confidence in Django more loudly than its blessing by Google.

Google engineers have been Python-centric for years (Python creator Guido van Rossum works at Google). But when Google launched AppEngine - a cloud-based web app framework - they released it with just one deployment option: A somewhat modified version of Django" [8].

	Django	Ruby on Rails	CakePHP
Language popularity	#6	#10	#4
Rating	5	3	5

Table 5: Popularity

6. Community and maturity

This measure focuses on the community and support for each framework, and how easy it is to get answers to questions that a developer might have. In other words, it shows how mature a framework is in terms of support, community, and documentation.

In the open source world where there is no official technical assistance 24/7, documentation and community support become crucial sources for developers. A well-supported framework is a big advantage. Documentation helps to build a stronger community, promotes discussion and improvement, and helps create momentum that leads to further popularity of the framework. Unfortunately, open source projects often lack well-structured documentation.

CakePHP documentation leaves a lot to be desired. The website is a little scattered. Documentation can be found in API, Docs, and Bakery. The documentation is well organized and well written but is light on content with some sections missing entirely for the new version of CakePHP (1.2). In general, the documentation is not bad, but sometimes it is hard to find what one is looking for because documentation is broken up into three groups. The community for CakePHP is not large but because this framework uses PHP it is usually not hard to find answers to programming related issues. It is harder to find answers to framework specific questions. However, a developer might find some useful information in one of the five books on CakePHP available on www.amazon.com.

Ruby on Rails and Django were extracted from web applications that were developed in the 2003-2004 period. Rails was released to the public in July 2004, and Django in July 2005. As such, Rails has had a head start in getting community contributions to the framework. There is a lot of documentation available. "Agile Web Development with Rails" is a book that became very popular between developers. A developer may choose between over 40 books written for Rails. Rails' community is mature and there is a lot of support available online, however some developers start feeling overwhelmed with the fast growing Ruby on Rails and are looking into more stripped down focused frameworks such as Django or Merb. To help solve this problem, Merb will be merged into Rails 3.

The Django website is very well organized and it is easy to navigate and get answers. All the documentation is contained within a Documentation folder. Documentation is further divided into such sections as the essential documentation, reference, deployment, and solving specific problems. Django takes documentation very seriously. Before the release of Django 1.0, a lot of documents were updated, and re-factored into small, easy to digest pieces. Django's documentation has a lot in common with the

> **"The Django website is very well organized and it is easy to navigate and get answers."**

Python documentation, and is one of the strengths of this framework. The community, however, is much smaller and less mature than the one for Ruby on Rails. One of the reasons for this is that Django was released a year later than Rails and 1.0 version was released only on September 3, 2008. There are about ten books available for Django from www.amazon.com. There is a lot of support for Python, since it is one of the most popular web programming languages.

Figure 5: Number of Books Available on Django, Ruby on Rails and CakePHP at http://www.amazon.com
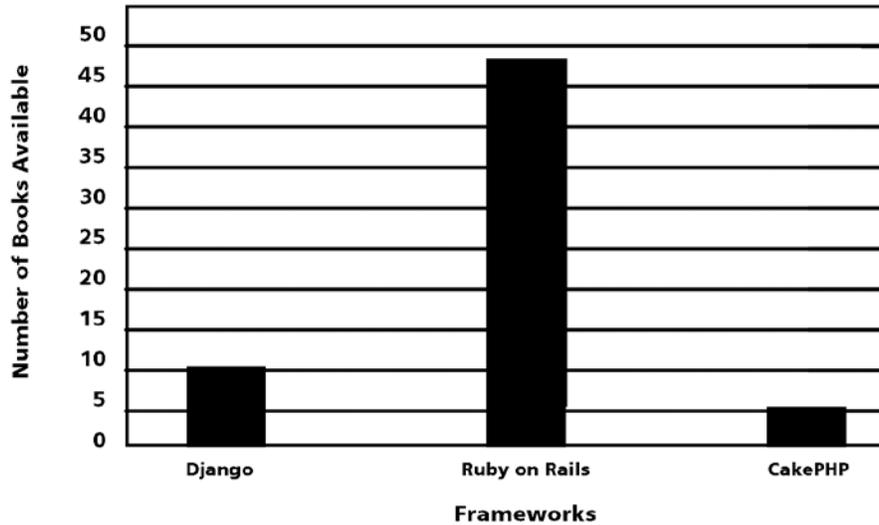
	Django	**Ruby on Rails**	**CakePHP**
API Reference	Good	Good	Good
Books	10	44	5
User manuals	Very good	Good	Poor
User blogs	Some	Many	Some
Community	Good	Mature	Small
Rating	5	5	3

Table 6: Community and maturity
**http://www.amazon.com**

7. Marketability

This measure focuses on how valuable it is to know a particular framework and the demand for developers. Marketability is directly related to the maturity measure. It also influences the popularity of a framework.

To evaluate marketability, the job market for these frameworks and programming languages were assessed. Indeed.com was used for this research because it includes sites as monster.com, careerbuilder.com and eliminates duplicate job postings. Jobs were assessed in San Francisco, CA, New York, NY, and Philadelphia, PA. The combined results for the three cities show that although there is a very low demand for CakePHP developers (about 17 positions), the demand for PHP programmers is very high (over 1200 positions). The demand for Django developers is a bit higher than the one for CakePHP developers (about 56). Similarly, the demand for Python programmers is pretty high as well (over 900). The situation for Ruby and Rails is a little bit different. There is a higher demand for Rails programmers (over 360) than for CakePHP and Django. However, the demand for Ruby (about 400) is lower than the one for PHP and Python. Figure 6 provides the graphic representation of the data.

Figure 6: Jobs on Indeed.com
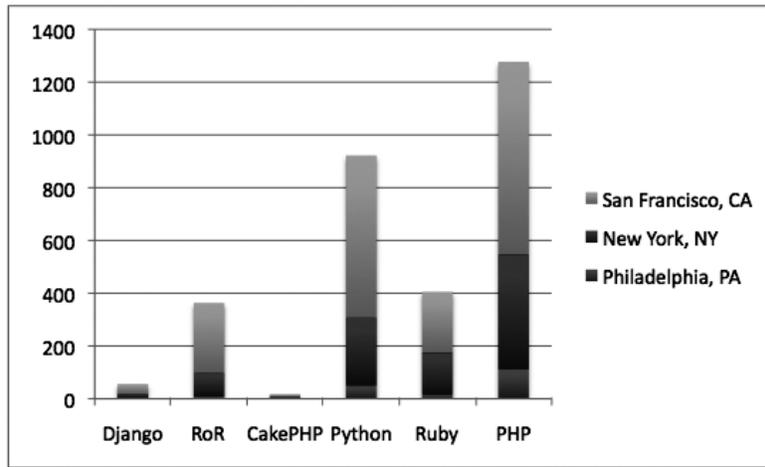
Figures 7 and 8 show job trends for the frameworks and the languages they use. The graph shows a high growth in Rails jobs, a slight growth in Django jobs, and yet a smaller growth for CakePHP jobs. Figure 8 shows increase in PHP positions, lower increase in Python positions, and yet a lower one in Ruby positions.

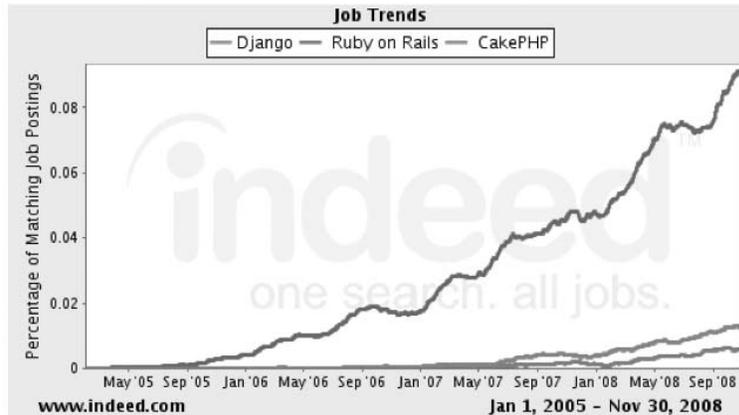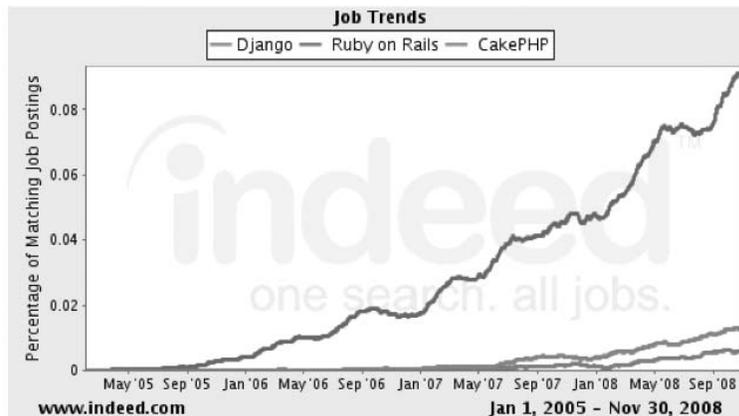**Figure 7: Job Trends for Django, Ruby on Rails, and CakePHP** (source: Indeed.com)

Figure 8: Job Trends for Python, Ruby, and PHP (source: Indeed.com)

	Django	Ruby on Rails	CakePHP
Current jobs and future trends (framework)	Slightly growing	Highly increasing	Low, but growing
Current jobs and future trends (programming language)	Medium demand	Lower demand	High demand
Rating	**4**	**4**	**4**

<div align="center">Table 7: Marketability</div>

Conclusion

Table 8 summarizes the results of the assessment. The weight of 0.2 was assigned to the criteria that measure the front-end and back-end components (user interface development, data management and migration). The lower weight of 0.15 was assigned to other metrics that analyze the overall performance of the frameworks (maintainability, testability). Other criteria (popularity, community and maturity, marketability) were weighed at 0.1.

> **"Django received the highest weighted score of 4.05.**
> **Ruby on Rails is second with 3.85 while CakePHP got 2.95."**

Django received the highest weighted score of 4.05. Ruby on Rails is second with 3.85 while CakePHP got 2.95. All three frameworks are powerful tools and the choice of which one to use will depend on context. Firms and developers should customize the evaluation method and weights assigned in this report to their unique context.

In general, if the IT staff of the company has extensive experience with PHP, CakePHP is a good choice, since it will kick start the development processes. Django or Ruby on Rails will take some time to learn, however in the long run, they will provide a more powerful underlying architecture and system that will support rapid application development. The main advantages of Django are automatically generated administrative interface and simple templates which can be used by non-programming web designers. The advantages of Rails are simple implementation of JavaScript, evolving schema, and large community.

Evaluation Criteria	Weight	Django		Ruby on Rails		CakePHP	
		Rating	Weighted Score	Rating	Weighted Score	Rating	Weighted Score
User interface development	0.2	4	0.8	3	0.6	3	0.6
Maintainability	0.15	4	0.6	3	0.45	3	0.45
Data management and migration	0.2	4	0.8	5	1	2	0.4
Testability	0.15	3	0.45	4	0.6	2	0.3
Popularity	0.1	5	0.5	3	0.3	5	0.5
Community and Maturity	0.1	5	0.5	5	0.5	3	0.3
Marketability	0.1	4	0.4	4	0.4	4	0.4
Total:	1		4.05		3.85		2.95

<div align="center">Table 8: Overall evaluation</div>

The focus of this report was on evaluating the internal attributes of each framework, it is however, just as important to consider how these frameworks will fit into an organizational software development cycle. The following provides conclusions on usage in an organizational context.

Software development practice and management

All three frameworks promote agile software development patterns and practices even in relation to tools and principles that are used for testing, deployment, etc. Developers within the frameworks' communities tend to use certain tools for source code control, continuous integration, automated testing and other tasks. As the popularity of the frameworks grows, there are more and more tools available for web developers. Git, Mercurial, Bazaar and Subversion are popular choices of version control tools; Fabric, Capistrano/Webistrano, Chef and Puppet are great deployment tools; RRDTool, Cacti, Graphite, Monit, Munin, Nagios, Zenoss are usage and availability monitoring tools; Integrity is a continuous integration server. The web frameworks do not provide tools for QA; however the support system can be set up. Given that Ruby on Rails is the most popular framework out of the three, there is a large community of developers surrounding it and as a result there is a wide spectrum of external tools available.

Scalability

Designing flexible web applications that are scalable is an important consideration for organizations. Applications may have to handle large spikes of traffic and data. There are ways to scale the applications in all three frameworks. As the demand increases, the application files can be moved to separate servers (media server, web server, database server). This process requires additional system administration, however the option is available and all three frameworks can be deployed within a clustered environment. Overall, since these frameworks are relatively new and often used for smaller web applications, it is too early to analyze their complete scalability. Large firms should consider using these frameworks initially for smaller or internal projects.

Role of open source frameworks

All three frameworks are widely used for small-scale project and are popular with start-ups because they are easy to use and learn, and can adapt very quickly to newer generations of tools. The benefits of ease of learning and adaptability follow from the open source community and "eco-system" that already exists for each framework. Enterprises, though, typically adopt new technology at a slower pace. Nevertheless, open source software is slowly becoming mainstream for enterprises. "The financial crisis and the threatening recession helped to promote Open Source even more and position it as a recipe to deal with lower IT budgets. Open Source adoption continued to be strong in Europe specifically. Germany, France and Spain alone can compete with the US in terms of adoption rates" [22].

Security considerations

Scripting languages bring security risks and some firms will not consider a toolset if the underlying code is scripted. These firms should consider JRuby, which is a "100% pure-Java implementation of the Ruby programming language" [11]. For a company that relies heavily on using Java, the existence of JRuby is the key selling point for giving the green light to the adoption of Ruby on Rails. JRuby compiles Ruby into Java-bytecodes and integrates with Java code and libraries. There is a Python implementation called Jython, and .Net developers can use IronPython. Further, Django is often used with the two popular open source databases, MySQL and PostgreSQL, but it works out of the box with Sun's Oracle, a popular database choice of larger corporations. There are various options for software stacks for each framework and the frameworks can often fit with a company's existing architecture. To limit initial risk, a company should considering evaluating usage in smaller projects, where the rapid development orientation of each framework can really shine and produce quick results.

References

Amazon.com. http://amazon.com.

Askins, Ben, Gree, Alan. "A Rails/Django Comparison." http://docs.google.com/View?docid=dcn8282p_1hg4sr9.

"Benefits of Using Open Source Software." June 2009. http://open-source.gbdirect.co.uk/migration/benefit.html.

CakePHP Cookbook. "Beginning with CakePHP." June 2009. http://book.cakephp.org/view/4/beginning-with-cakephp.

CakePHP Official Website. June 2009. http://cakephp.org.

Croft, Jeff. "Django Templates: An Introduction." February, 21, 2006. http://jeffcroft.com/blog/2006/feb/21/django-templates-an-introduction.

Django Official Website. June 2009. http://www.django-project.com.

Hacker, Scot. "Notes on a Django Migration." November 19, 2008. http://birdhouse.org/blog/2008/11/19/notes-on-a-django-migration.

Holovaty, Adrian, Kaplan-Moss, Jacob. "The Django Book." 1st edition. http://www.djangobook.com/en/1.0.

Indeed. Job Search Engine. http://indeed.com.

JRuby. July 2009. http://jruby.codehaus.org.

Kaplan-Moss, Jacob. "Becoming Open Source Developer." PyCon 2007. Python Software Foundation.

Lily, Justin. "Django at 30,000ft. A Manager's View." Python Magazine. Vol. 2 – Issue 9. September 2008.

Model – View – Controller. Wikipedia. http://en.wikipedia.org/wiki/Model_view_controller.

"MVC: The Most Vexing Conundrum." Slash 7. February 2, 2005. http://slash7.com/articles/2005/2/22/mvc-the-most-vexing-conundrum.

PHP Programming Language Official Website. http://www.php.net.

Python Programming Language Official Website. June 2009. http://python.org.

Ruby on Rails Official Website. June 2009. http://rubyon-rails.org.

Ruby Programming Language Official Website. December 2008. http://www.ruby-lang.org.

Slater, Michael. "Can Rails Scale? Absolutely!" Building WebApps. April 5, 2008. http://www.buildingwebapps.com/articles/6419-can-rails-scale-absolutely.

"TIOBE Programming Community Index." TIOBE Software. http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html.

Von Rotz, Bruno. "Open Source Year 2008 in Review: More Adoption, Success, Innovation, and Alternatives". Optaros. December 21, 2008. http://www.optaros.com/blogs/open-source-year-2008-review-more-adoption-success-innovation-and-alternatives.

Willison, Simon. "An Introduction to Django." London Web Frameworks Evening. November 17, 2005. http://simon.incutio.com/slides/2005/lwfe/IntroductionDjango.pdf

About the author

Julia Plekhanova is a web developer, web designer and a co-founder of the freelance collective Insatsu. Her interests include open source technology, agile web development, interface design and data visualization. In 2001 Julia was awarded the highest honors in her graduating class for outstanding achievements in learning. In 2006 she was named to the dean's list for the spring semester. While studying at Temple University, Julia was a web master for the Association of Management Information Systems (AMIS).

Julia's academic foundation in computer technology and management supplied her with a strong foundation in application development and project management. She is enthusiastic about developing web applications, and takes great care every step of the way to pay attention to detail, while focusing on the solution. Julia is passionate about learning new technologies and perfecting her skills. She believes that challenges are the best tools for learning and achieving new goals.
Julia holds a BBA in Management Information Systems from Temple University. 'Evaluating Web Frameworks' report is Julia's first publication.

Contact information:

Julia Plekhanova
PO Box 306
El Prado, NM 87529, USA
yulka@insatsu.us

Fox School of Business and Management

Established in 1918, Temple University's Fox School of Business is the largest, most comprehensive business school in the greater Philadelphia region and among the largest in the world, with more than 6,000 students, 150 faculty, and 51,000 alumni. In 2008, *Academic Analytics* ranked Fox School management information systems faculty in the top 10 for research productivity while *TechRepublic.com* selected the Fox School's bachelors degree in information systems as one of the top ten programs in the U.S. The Fox School's programs are among the best in the world and are highly ranked by the *Financial Times*, *The Economist*, *U.S. News and World Reports*, *Princeton Review*, and *Computerworld*.

Institute for Business and Information Technology

The Fox School's Institute for Business and Information Technology (IBIT) provides the cutting-edge knowledge and people to create and sustain excellence in information technology. IBIT offers participating corporations a membership structure so that they can leverage and influence our knowledge, human capital, and established network. IBIT leverages The Fox School's research expertise, educational resources, global presence, and entrepreneurial spirit to prepare business leaders and create industry relevant knowledge. IBIT offers the following programs:

Fox IT Symposium
The Fox IT symposium is an exclusive highly interactive forum of
noted practitioners addressing current topics.

Distinguished Speaker Series
The series features talks by leading professionals on important business technology topics.

Fox IT Awards
The Fox IT Innovator, Leader, and Distinguished Alumni awards are presented to i
ndustry leaders at the IT Awards Reception.

Research
IBIT affiliated researchers publish in top academic journals, conduct workshops, and release
The IBIT Report on important industry relevant topics.

Scholarships and Fellowships
IBIT recognizes exceptional students with scholarships and awards.

Industry Projects
IBIT affiliated faculty and students work with local organizations on
joint business technology projects.

Workshops and Special Events
IBIT organizes special workshops and forums that allow academics and
industry leaders to exchange ideas and produce knowledge.

Web2.0-Frameworks dagegen erheben den Anspruch der schnelleren und einfacheren Entwicklung. Exemplarisch wird in diesem Schlagwortbeitrag das neuartige Web2.0-Framework Ruby on Rails (RoR) vorgestellt. RoR versucht die einfache U View. Django Application Management. Chapter. Oct 2017. Daniel Rubio. Django, like all modern application development frameworks, requires that you eventually manage tasks to support the core operation of a project. This can range from efficiently setting up a Django application to run in the real world, to managing an application's static resources (e.g., CSS, JavaScript, image files). View. Not finding the right answers on Google? Mojolicious, Ruby on Rails, Django, Zend Framework, Yii, CakePHP and. Django 1.0 Template Development - Packt Publishing. Frameworks for Model-Driven Development of Web Applications. This white paper is a technical manager's guide to evaluating web development frameworks, with a detailed review of Ruby on Rails and the Django (Python) projects. In this study we establish a list of evaluation criteria and then discuss and rate how well each framework addresses each criteria. As your priorities and assessment may differ from ours, this report can be used as a road map for your own evaluation. The list of top web development frameworks changes with the speed of lightning. Angular and Node, Vue.js, Symphony and CakePHP – there are tons of advanced tools dedicated to the web development. These tools have gained amazing popularity and are considered to be among the best frameworks for the web development. Recommended : Python VS Ruby comparison. Ruby on Rails vs Django: which performs better? Which one to choose? We're going to have to take a rather long journey to answer these questions. RoR and Django: the Similarities. Python with Django is slower than Ruby on Rails. Raw language or framework speed can be tested only by measuring it via the real experiment on a real app.

I was thus looking at CakePHP and Ruby on Rails as frameworks. However, I am not familiar with either of these technologies (I do have a software background, but it is mostly C/C++/Java). So which do you think would be best for me to pick up for this project that will enable me to quickly and easily just 'build' something like this for Facebook? Having developed in both CakePHP and RoR - if you're coming with no web development background and you're looking to start; dive in with either. Honestly, it'll be the same learning curve for you! You will find the setup, learning, development and deployment easier in CakePHP - PHP is one of the most popular languages.