

WHITE PAPER

New Product Development with Ruby on Rails

by Selina D'Souza

There are huge changes in the way software is being built today and the timeframes in which it gets built. The reasons for these changes are manifold and have deep implications for software companies. Ruby-on-Rails supports rapid development, extensive collaboration and can be a great platform of choice for building new web applications and products successfully.

Table of Contents

1. The Changing Rules for Web Startups.....	2
2. Ruby on Rails (RoR) – Platform for New Product Development.....	2
3. Advantages of the Rails Framework	3
4. Web 2.0 and Rich Internet Applications.....	3
5. Ruby - Dynamic and Elegant.....	4
6. Behavior Driven Development with RoR.....	4
7. Shortcomings of Rails.....	4
8. Hosting Rails.....	5
9. Conclusion.....	5
10. About Aspire.....	6



New Product Development with Ruby on Rails

The Changing Rules for Web Startups

The early 2000s (post-dotcom crash era) had VCs investing cautiously in technology start-ups. Hardly five to six years later, VCs are having a hard time finding software companies that need millions of investment dollars upfront. Internet start-ups no longer keep themselves in stealth mode for years on end, and don't need the heavy investment that was needed earlier to just get off the ground.

The reasons for this change are manifold. Hardware costs have reduced dramatically, cloud computing has made web infrastructure a trivial issue, software libraries are ubiquitous and new frameworks have crashed development timeframes. Whereas, earlier a web startup would typically need a seed investment of at least a million dollars, today internet start-ups comfortably take a product to the market with seed investments of \$50-500K in just a period of 4-6 months.

Another major reason for this revolution in software development is the advent of Web 2.0 and social media. Users have become more participative in the whole development process and have become contributors and co-creators, rather than mere end-users. They are willing to try products in beta mode much more and much earlier than ever before.

This change in the way software is being built and the timeframes in which it is getting built has deep implications for software companies and development teams. They need to find a way to adapt to this change – to build better products with the new technology that is available, incorporate users' feedback consistently, and do all this in ultra fast development cycles.

Ruby on Rails (RoR) – Platform for New Product Development

For startups that have a brilliant idea and are looking at taking their concept to launch quickly, Ruby on Rails (RoR) can provide a platform to build new web products easily. RoR is an open-source web application stack, comprising of the Rails framework, which has been developed using a pure object-oriented language called Ruby.

Rails has a wealth of pre-written code that implements the structure and many of the common functions of a database-driven site. Also, RoR gives you the power to innovate and iterate as it supports Agile development and can help you create working software rapidly.

While there are other similar frameworks like PHPCake, CodeIgnitor, Trails, and dozens of others (each with their own bandwagon of supporters and critics), Ruby-on-Rails makes common web development tasks 'fall-down-the-stairs' easy. It eliminates much of the preliminary work necessary to create complex websites and database-driven products.

RoR allows generation of web pages on the spot, includes advanced Ajax support for a dynamic online experience, and has many popular components like ORM, Web services, JSON and other APIs built-in. These components can integrate with external components and reduces development time and effort significantly.

While RoR is perceived as a good framework for rapid prototyping and pilot web applications, but not for heavy-weight development, in reality, Rails enforces a tight, organized structure to the code. This ensures maintainability and allows for future scaling.



New Product Development with Ruby on Rails

Rails also uses a Model-View-Controller style of architecture, in which the database and its representation ("model"), the user interface ("view"), and the functional logic ("controller") of the application are all kept neatly separate. This architecture style typically helps create products that are clean, portable and sharable.

Advantages of the Rails Framework

Rails has a powerful tool in its Active Record Model which allows direct communication with the database layer and requires zero configuration for Object Relational Mapping and reflection. This allows developers to deal directly with objects, even if they are actually dealing with XML, relational database data, or Javascript in the browser.

Rails' multi-view development is another big advantage for web startups and any online company for that matter. What this means is that with the same back-end code base, a web page can be rendered in different views or output formats like HTML for a normal browser, RSS for an RSS reader or in a format that can be viewed on an iPhone. This can allow you to extend your product to multiple devices. While this was possible earlier, the latest version of Rails provides new features that make this type of cross-platform development much simpler.

Rails supports just about any database and permits easy database portability. One of the attractions of Rails is its scaffolding, which makes database tables editable on the web in mere seconds, and with just a few lines of generated Ruby code. It provides a simple interface for dealing with data while it does all the hard work of form generation and handling and even provides clever form generation to support different input types (like simple text strings, textareas, date selectors, datetime selectors etc.)

Rails also has a migration framework that lets you perform or revoke changes in multiple databases in a fraction of seconds. Other key Rails highlights are its integrated Test environment and Auto-deployment.

The integrated Test environment makes testing in Rails simple and well-documented. Rails builds the necessary scaffolding for unit and functional tests, and tracks test/code ratios automatically. It includes native support for mock object testing and database fixtures, streamlining out-of-container testing. This availability of ready testing catches quite a number of defects early in the development cycle, when it can be rectified easily.

Web 2.0 and Rich Internet Applications

As it is to be expected from a web application development framework, Rails has abundant Web 2.0 and rich internet application support. Ajax (it allows a part of a webpage to be updated with new information without the need to refresh the entire page) which has revolutionized traditional web development in recent years by making it more responsive and dynamic, has built-in support in RoR.

Infact, Rails makes Ajax so easy, that for typical cases, it's no harder to use Ajax than it is not to! And this is without having to deal with the idiosyncrasies of different browsers. Rails even includes seamless integration with a couple of animation and Javascript frameworks (such as script.aculo.us and Prototype) that have cross-browser UI Ajax and Javascript libraries and tool-kits to ease dynamic web development.

Adobe Flex and QTRuby are other tools that RoR supports to create rich internet applications. RoR also supports a lot of opensource IDEs like RadRails, NetBeans and Aptana Studio and database tools like HeidiSQL and others to quicken development.



New Product Development with Ruby on Rails

Ruby - Dynamic and Elegant

One of the surprises about Ruby on Rails is the Ruby language itself. Although it had been around for more than a decade, Ruby did not have many takers until it was used to create the Rails framework. Ruby is a dynamic language (it isn't compiled) and is more akin to Perl and Javascript than a traditional hardcore language like Java or C++. But that's where similarities to Javascript end since Ruby is an elegant object-oriented language designed to make programming easy and make routine software development tasks almost effortless.

Although opinions differ vastly across the board, RoR evangelists vouch for Ruby in terms of the development time, compactness and readable code generated compared to other similar languages. More than anything though, RoR is perceived to be fun to work with (compared to traditional frameworks), and one of the reasons for this is the Ruby language itself. That, by itself, can be a motivation for some developers.

Behavior Driven Development with RoR

Another aspect of Ruby-on-Rails that is supportive of Agile development methodologies is the availability of RSpec, a Behavior Driven Framework for Ruby. BDD encourages collaboration among all stakeholders in a product development effort. It encourages developers to think of the behavior of the component that they are developing, and of the roles and responsibilities of the other objects it interacts with.

With BDD, testing becomes less about writing tests, and more about writing specifications and defining the behavior of the program being developed. RSpec provides two frameworks (a Story and a Spec framework) for writing and executing examples of how a Ruby application should behave at the application and at the object level. This allows expression of stories with different scenarios and then writing tests to make sure the scenario passes. The logic of the code being written can be tested right away with the story and spec framework.

Shortcomings of Rails

While Rails has a lot going for it in terms of the ease with which it can create web products, it is probably not the best bet for simple website development and products that are not heavily database-driven.

Also, Rails' structured and organized approach can be its strength as well as its weakness in terms of the flexibility that it loses by enforcing strict rules within the confines of the framework. And while RoR is a platform that is gaining acceptance among more and more web developers, it hasn't reached the maturity of say a Java-based framework like J2EE, which has been around for many years.

Scalability has been a criticism often thrown at Ruby on Rails. While there will certainly be glitches when there is unprecedented, explosive growth (say from having a few hundred users to millions of users within a few weeks), one must remember that any other database-driven web platform will face the same bottle-necks that RoR faces when scaling to such a large extent in a short timeframe.

At the end of the day, RoR or any framework for that matter is not the right choice for every single type of development or situation. Based on the nature of the project and product being developed and its precise requirements, the right technology and tools for the job need to be judiciously chosen.



New Product Development with Ruby on Rails

Hosting Rails

For start-ups that are making a decision to move or have already moved towards RoR as a development platform, the availability of a number of exclusive RoR hosting providers is something that they can leverage to their advantage. Without having to worry about a large capital layout for web infrastructure, deployment issues or trained IT staff to manage their fledgling RoR web product, startups can concentrate on understanding their customers better, fine-tuning and marketing their product.

RoR application hosting vendors offer different plans for different needs – including pre-installed software and hardware, 24/7 support, the ability to scale from hundreds of page views to literally billions of page views a day, shared or dedicated servers, and importantly payment for only what is used. Deploying, running and hosting a product in the cloud has never been easier!

Conclusion

Ideas and imagination are key to creating innovative new products and services today - infrastructure and other setup issues are no longer a barrier. At a recent event discussing startups and innovation, Tom Duterme from Google's New Business Development group, mentioned that there are three things that startups should make use of in this internet age: hammers (tools), wires (broadband) and rucksacks (storage).

Apparently, Google looks for acquisitions that take advantage of the above things. Tom next went on to give this advice to startups in order to commercialize their ideas, which seems quite pertinent:

- Collaborate and get the right people on your team.
- Fulfill user needs: 70-80% of ideas that fail do so because of lack of user focus.
- Iterate often ("Big will not beat small anymore. It will be fast beating slow"; quote from Rupert Murdoch)

Ruby-on-Rails, with its support for rapid development, extensive collaboration through Behavior Driven Development and Agile methodologies, and a Web 2.0 focus can certainly be a good platform of choice to help build new products and services for the next generation.



New Product Development with Ruby on Rails

ABOUT ASPIRE SYSTEMS

Aspire Systems is an Outsourced Product Development firm committed to helping our customers build software products better and faster. We work with some of the world's most innovative Independent Software Vendors and software-enabled businesses, ranging from start-ups to established industry leaders, transforming the way software is built.

Aspire provides complete product lifecycle services, ranging from new product development and product advancement to product migration, re-engineering, sustenance and support. Our product development teams are spread between our Global Innovation Center in Chennai, India and offices in the United States.



Aspire Systems India Private Limited
Plot No 1/D-1, SIPCOT IT PARK, Siruseri, Tamil Nadu - 603 103
Tel : +91-44-67404000. Fax: +91-44-67404234
E-mail : info@aspiresys.com
Web: www.aspiresys.com