Ruby on Rails – An Agile Developer's Framework

S. Meenakshi
Associate Professor
Department of Computer Applications,
R.M.K. Engineering College, Kavaraipettai

ABSTRACT
Agile development framework is a free, open source web application development framework. It aims at expanding the speed and ease with which database-driven web applications can be made. Agile web development emphasizes clear goals, planning, incremental and iterative delivery in this way guarantees the successful completion of a product at the end of each iteration. Successful interaction increases success of the application because of the core focus mainly on client participation. This declines problems that come with the development team changing the product in a late stage due to changes in requirements. Rather, both the development team and the clients are informed, therefore client confidence remains high and there are minimal delays till the end of the development and in addition deployment. All tasks are performed at defined period, with simply less documentation work. The agile environment deal with rapid application development and in this manner making developers to do software and web solutions in a shorter timeline[1]. Organization ranging from start-ups to established enterprises and have arrived at the conclusion that when making web applications, Ruby on Rails is the best choice. This paper highlights the Rails framework characteristics and its relevance to adaptive nature, iterative and incremental development using agile methodologies[2].

Keywords
Ruby on Rails (RoR), Rails Framework, Agile development, MVC architecture.

1. INTRODUCTION
In the recent years there has been a rising of a new style of software framework being referred to as agile-rails framework [3]. Since 1990s it developed out of the efforts of many peoples who dealt with software process [8], discovered them having, furthermore anticipated for a new framework to web development. Ruby on Rails is an open source web based application framework composed using ruby programming language [5]. It is focused on Model-View-Controller architecture. Rails framework is sufficiently basic to require just a couple of lines of code to raise a whole web application with less configuration [7].

1.1 Ruby (The Programming Language)
"Ruby is a object-oriented programming language that is translated. Ruby started during the mid-1990s in Japan and was initially developed and designed by Yukihiro. It has syntax similar to Perl and semantics to Smalltalk. Ruby itself was developed in C dialect. Ruby was made with less developer work and also possible confusion. In light of these reasons, the Rails framework was written in ruby. Ruby has started to become popular worldwide in the recent years.

1.2 Rails (The Framework)
Rails are a web framework built on Ruby, consequently the name Ruby on Rails. It empowers the programmer to easily create progressed database-driven websites using scaffolding and code generation, taking convention over configuration, which means that if you adhere to a certain set of conventions, a lot of features will work right out of the box with almost little code. Rails are an open source Ruby framework for creating web applications.

1.3 Ruby on Rails (RoR)
Ruby on Rails (RoR) is a web framework written in Ruby. Ruby on Rails makes it simple to assemble a database-backend sustains web application that uses the language Ruby.

1.4 Outline of Paper
Section II presents architectural design outline of the agile and RoR. Section III presents how RoR empower agility. Section IV presents developer's view: Rails Vs J2EE Framework. Sectional V presents a experimental results and Finally Section VI will finish up with a discussion of future work.

2. ARCHITECUTRE
2.1. Agile Design Architecture
Agile design architecture involves gathering of software development methodologies based on iterative and incremental development, wherein requirements and solutions go ahead through collaboration between a self and cross-functional team [9].

Agile methods break major tasks into smaller number of sub-tasks called increments with minimal planning, and don't specifically involve long-term planning. Iterations are shorter timeline that ordinarily takes from one to four weeks at the maximum [10]. Each iteration make use of team working through a full development cycle which incorporates: planning, requirements analysis, design, coding, unit testing, and acceptance testing when a working product is at last shown to stakeholders. This helps to minimize overall risk, and lets the project adjust to changes quickly. Iteration may not add enough functionality in order to warrant a business sector release, but the objective is to have an accessible release with as much as minimum bugs at the end of every iterations [17]. Multiple iterations may be required to release a product or features [20].

The below figure demonstrates the steps in Agile design architecture which concentrate on iteration and adaptable to change.

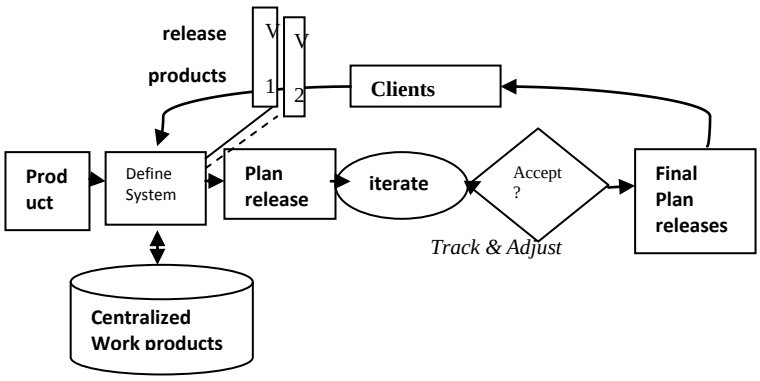

Figure 1. Agile Design Architecture

2.2. Rails Design Architecture

Rails are focused on a programming pattern called MVC, which remains for Model-View-Controller. The Goal of Rail design outline is to discrete organization of data (model) from UI & presentation (view) by introducing controller.

The MVC Architecture

One of the important feature of Ruby on Rails is it depend on Model-View-Controller architecture (MVC)[22]. The main advantage of MVC is to have the detachment of Business logic from client-user interface. Other advantage incorporates ease of keeping code, DRY and making it clear where distinctive sorts of code belong for less demanding upkeep.

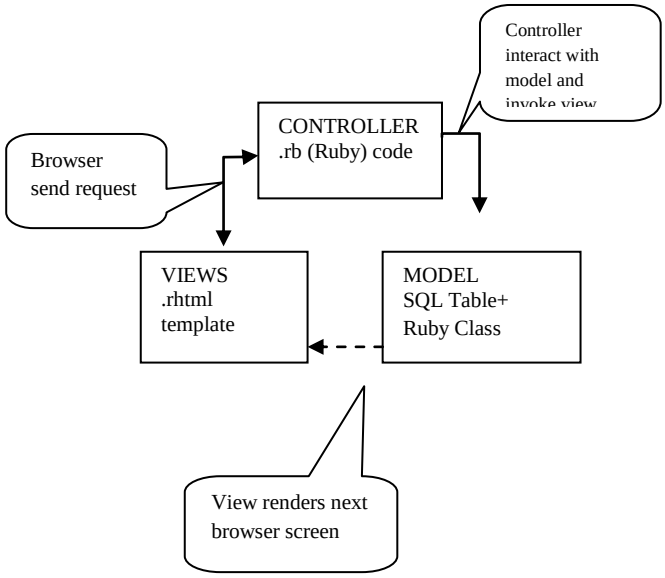

Figure 2. MVC Architecture

Model: Model stores the information (data) about each table of database. If there should arise an occurrence of Rails, do some primary validation of data before putting into database and one table relate to one model.

View: View, for the most part the interface of the application. If there should be an occurrence of Rails, Views are often HTML pages with some inserted ruby code. Views handle the job of giving the data to web browser.

Controller: Controllers gives the "glue" in the middle models and views. In Rails, Controller fundamentally handles the incoming request from the web browser interrogating the models for data and passing that data on to the views for presentation.

2.3 The Components of Rails

Architecture of Rails framework is illustrated in Figure 3. Rails architecture comprises of two noteworthy elements which incorporates Active Record and Action Pack.

Action Record is an Object Relational Mapping (ORM) layer which handles the Model element of the MVC application. This layer covers the data table rows as model objects to the rest of the application.

Action Pack handles the view and controller elements of the MVC application. Action Pack comprises of two parts viz. Action View and Action Controller which handle View and Controller of MVC application individually.

Each of the architectural elements in the picture above is detailed below:

(a) User component – Rails web application is accessed by browsers using different protocol interfaces such as HTTP, FTP or expended as a web services using SOAP.

(b) Web Server – The developed Rails application dwells on a Web Server, which handles request from the client-user component and advances them to the Dispatcher.

(c) Dispatcher – Dispatcher invokes proper controller focused on the current user request.

(d)Controller –Controller processes the user request and invokes proper action.

The action works together with appropriate model objects and prepares the response. In the event that the request was from the browser, the response will be rendered through Action View Component else a suitable response is relayed by either the Action Web services or the Action Mailer Component.

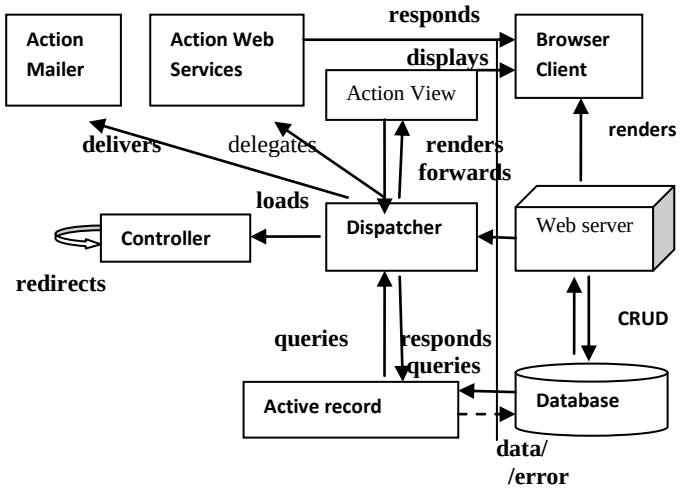

Figure 3. Architecture Model for Ruby on Rails (RoR) Web Application Implementation

3. HOW DOES RUBY ON RAILS ENABLE AGILE DEVELOPMENT

What makes rails agility which thus infers how Ruby on Rails empowers agile development? In this section might elaborate the characteristics of Rails framework that make it most appropriate for agile methodologies.

MVC architecture - which permits web professionals to partition business logic, user presentation and control features and encourage the process of web development.

CRUD (Create, Read, Update, Delete) - In the heart of the Rails routing engine lies CRUD, **or** Create**,** Read**,** Update**,** Delete, which originates from a notion that all web pages are comprised of these four actions. For example, can make a product, view, or read a client-user, edit, or update a category, and delete an order.

If follow this pattern, will accomplish a great deal of work done. For example, all it takes to make all of these actions for a product is to make the routing engine that have an entity called product, and it will wire up all the routing logic that ties the request to the matching controller.

DRY – "Don't Repeat Yourself" – suggests that writing the same code over and over again is a terrible thing.

Convention Over Configuration – implies that Rails makes presumptions about what need to do and how are going to do it, rather than letting to change every little thing through interminable configuration files.

REST (pattern) - is the best pattern for web applications for sorting the application around resources and standard HTTP is the fastest approach to go.

Scaffolding - often make transitory code in the early phases of development to help get an application up rapidly and perceive how major components work together. Rails automatically make a great part of the needed scaffolding.

Portable programming skills – Team managers will admire the way that Rails minimizes configuration and encourages standardization. This empowers the programming skills more portable.

Rapid feedback loop – In Rails yet get almost-instant feedback as with code. Thus consequences of the progress made can be immediately seen improving the overall client-customer experience empowering changes made instantly as well.

A great number of software development companies strive to make the procedure of web application development less tedious and more flexible; hence, such organizations and its web specialists take advantage of Ruby on Rails technology, which allows them to utilize the principles of agile development.

4. DEVELOPER'S VIEW: RAILS VERSUS J2EE WEB FRAMEWORK

Considered as agile web frameworks and Rails won over Java Web Framework as well. Unfortunately, other frameworks are not in this comparison. In perspective with Developers, the ruby has the following advantages and comparison with J2EE framework:

4.1 Ruby Advantages

Contrasted to Java, Ruby is a much more adaptable language. Map and inject methods permit the code to be composed in a compact and readable way compared to the conventional way of wrapping the code inside loops through passing in Ruby code blocks.

Strings in Ruby are mutable and provide a extremely comprehensive API. More compact syntax and some more adaptable classes allows numerous tasks to be performed much more rapidly in Ruby than in Java. Having to deal with less code also prompts to a much smaller, more viable codebase.

A little, better maintained codebase leads to a more complete test scope and less wasted time in refactoring, all prompting to a much more productive and enjoyable software development.

4.2 Rails versus J2EE Web Framework

The Ruby on Rails is most likely the most comparable framework to a usual J2EE stack. Many of the libraries found in J2EE have a counterpart in one form or an alternate in Rails. Ruby counterparts are frequently less powerful and adaptable; however require less maintaining and less code from a developer to perform the same task.

Rails is significantly more invested in tradition than in configuration. J2EE libraries like Hibernate and Spring are very adaptable and configurable however accompanied an increased cost in maintainability.

Both Spring and Hibernate gives an extremely adaptable structure through effortlessly replaceable modules. Hibernate gives an experienced developer a great deal more control on how the application flow will be processed.

Spring offers a lot of different modules for complete customization, giving quite a bit of common functionalities for a web application.

Rails accompany a lot of handy and powerful generators that make it extremely easy to take after the set of conventions.

Setting up a project quickly from scratch makes building RESTful applications in MVC architecture with Ruby on Rails is requests of extent faster than with its J2EE counterparts.

Maintenance - Ruby counterparts are often less effective and adaptable, yet do also require less maintaining and less code from a developer to perform the same task.

Standard Convention - Rails is significantly more resource into tradition than in configuration. Rails accompany a lot of handy and powerful generators that make it to a great degree simple to follow the set of conventions.

RESTful Application - Setting up a project rapidly from scratch makes building RESTful applications in MVC architecture with Ruby on Rails is orders of magnitude quicker than with its J2EE counterparts.

5. EXPERIMENTAL RESULTS

"Ruby on Rails is a breakthrough in lowering the barriers of entry to programming. Powerful web applications that formerly might have taken weeks or months to develop can be produced in a matter of days."

-Tim O'Reilly, Founder of O'Reilly Media

In this case study, created a support web log application using Ruby on Rails and Net Beans. The steps are given as follows:

Step1: In Net beans, Start by creating a new project| Ruby| Ruby on Rails Application| project Name-weblog.

This will create all the files and dirs. needed by a standard project including Controllers, models and views

Step2: Select Controllers – look for db configuration where using MYSQL database by default database as weblog development. Hence by outside netbeans in MYSQL commands line client - MYSQL > create database weblog development;

Step 3: Now ready to create a model, use generator to generate model

Select weblog | generate | model

In arguments – entry title: string content: text

Model will be called entry – one entry for each for weblog entry

And there will be two fields: title and content

Jreader creates DB migrations scripts, this script will run to create

the table – that's the model

Can see the migration changed for the two columns tile and the content

Step 4: In weblog|Database Migrations|Migrate|001_create_entries.rb

If def is self.up – create the table

If def self.down – if reverse the migration the table will be

dropped.

Step 5: Let's apply this migration now.

Weblog| Migrate Database |To current version

Once migrated now ready to create the fields and controls

needed to create an application

Step 6: Use the generator to create scaffold for the model - entry

Step 7: At this point completed the application so as to run

Weblog|Run

It has been switched to http://localhost:3000/entries and created a

new one

Step 8: Listing Entries- Title – Content - New Entry click on

Step 9: In Title – Hello world!

In Content – This is my first entry. Yeah press create button.

As of right now effectively created first weblog application using Ruby on Rails and Netbeans in a moment.

6. CONCLUSION AND FUTURE ENHANCEMENTS

Ruby on Rails as an open source web application development framework focused around Ruby programming language and used extensively by agile programmer teams has been pretty popular for rapid web application development. Two interesting aspects of RoR stand out – the 'convention over configuration' wherein programmers specify the unusual configuration as well as 'Don't Repeat Yourself' wherein information used is unique and not duplicated.

Rails are a perfect platform for Agile development practices. Keeping in view its agreement with the Agile Manifesto, it is not difficult to foresee that Rails will be the future of Agile development[11]. As a fairly new web framework, Ruby on Rails has not yet seen the prominence of a number of today's current standards such as Java or .NET based frameworks[12]. This does not imply that Ruby on Rails is unequipped of handling real world applications, as it can and has been taking care of wide range of dynamic web-support-projects [15]. Ruby on Rails is a extraordinary framework for creating progressed web applications writing very little code in comparison [16]. Recommend it for almost any kind of application, and particularly for prototyping [19]. To have a firm conclusion that a blog can be created within a moment using Ruby on Rails. Due to its straightforward nature and simple to use features as well as the speed of project completion,

7. ACKNOWLEDGMENT

Would like to thank the institution, referred to in this paper anonymously, without which this piece of research would not have been possible. Also like to thank the many faculties within the Institution for their cooperation and support. Grateful to thank HOD-Computer Applications of the Institution, for the constant support given by him, and thank the unknown referees for their helpful comments and suggestions to improve this work.

8. REFERENCES

[1] Nerur S, Mahapatra R & Mangalraj G (2005), Challenges of migrating to agile methodologies Communication of the ACM, Vol. 48 No. 5, Pages 72-78 New York USA.

[2] Martin Fowler "New Methodology" , http://martinfowler .com/ articles/ newMethodology.html Vol. 1 de 17 (19/08/2010)

[3] Andrew Mcdonald, "Agile Web Engineering (AWE) Process: Perceptions Within A Fortune 500 Financial Services Company" Journal Of Web Engineering, Vol. 4, No.4 (2005) 283-312.

[4] http://en.wikipedia.org/wiki/Agile_software_development.

[5] Osmar Zaine, "Report on Ruby on Rails", CMPUT 410, November 16, 2007.

[6] Bram Wiebe, Trevor Jones, Wanying Luo, Howard Chung -group presentation, "Ruby on Rails**"**, Osmar Zaine November 16 2007.

[7] Qumer., & Sellers, An evaluation of the degree of agility in six agile methods and its implacability for method engineering. Information and Software Technology, (2008) 280-295.

[8] Phalnikar, Deshpand, &Joshi, Applying Agile Principles for Distributed Software Development - International Conference on Advances in Computing and Communications 01/2009; DOI:10.1109/ICACC.2009.93 ISBN: 978-0-7695-3516-6.

[9] Judy, Simon & Schuster, Agile principles and ethical conduct. Proceedings of the 42nd Hawaii International Conference on System Sciences-2009.

[10] Malik Hneif, Siew Hock Ow, "Review Of Agile Methodologies In Software Development", International Journal of Research and Reviews in Applied Sciences ISSN: 2076-734X, EISSN: 2076-7366 Volume 1, Issue 1(October 2009).

[11] Asad Awan, "Agile Web Development with Rails – An Overview", Research article Friday 16. of January 2009.

[12] Sudipto Das, Vinod Kone "Ruby under Scanner: Comparison with Java" University of California, Santa Barbara 2010

[13] João M. Fernandes and Mauro Almeida, "A Technique to Classify and Compare Agile Methods" (Eds.): XP 2010,

LNBIP 48, pp. 385–386, 2010. Springer-Verlag Berlin Heidelberg.

[14] Dwayne Read, "Going Agile- A Case Study", 2010

[15] Aspire Systems, "Ruby on Rais-Building Dynamic Web Products Faster", 2011.

[16] Feng Ji and Sedano T, "Comparing Extreme Programming and Waterfall Project Results" Carnegie Mellon University June 2011.

[17] Nystrom Anna, Agile Solo, "Defining and Evaluating an Agile Software Development Process for a single Software Developer" Chalmers University of Technology June 2011.

[18] S.Balaji, Dr.M.Sundararajan, Murugaiyan, "Wateerfall Vs V-Model Vs Agile: A Comparative Study On SDLC", International Journal of Information Technology and Business Management 29th June 2012. Vol.2 No. 1 2012.

[19] Brian Wernham, "Agile Project Management for Government Case study: The Success of the FBI Sentinel Project" , Talk given at the Agile Business Conference (ABC2012) 10th October 2012 held at the Inmarsat Centre, London

[20] Mr. D. T. T. Vijaya Kumar & Ms. M. Sowmyavani, "Agile Software Development: A Case Study of Web Application" , Global Journal of Computer Science and Technology Software and Data Enbgineering, Volume 12 Issue 14 Version 1.0 Year 2012.

[21] Bram Wiebe, Trevor Jones, Wanying Luo, Howard Chung, Kaushal Pathak Anju Saha, Review of Agile Software Development Methodologies, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 2, February 2013 ISSN: 2277 128.

[22] Sam Ruby, Dave Thomas, David Heinemeier Hansson, "Agile Web Development with Rails", Third Edition, 2013.

9. AUTHOR'S PROFILE

S. Meenakshi received Master of Computer Applications under Madras University, Chennai and Master of Philosophy from Periyar University, Erode. Currently she is working as an Associate Professor in the Department of Computer Applications, R.M.K. Engineering College, Kavaraipettai and pursuing Ph.D. programme in the Department of Computer Science, Mother Teresa Women's University, Kodaikanal.

Prior to join in RMKEC she was working with web since 2003. She has been an author for many web enabled projects. She also presented many papers in National and International conferences and Journals. She has published a Book on Web Graphics, Scitech Publications, Chennai. He is a life member of ISTE (Indian Society for Technical Education) and CSI (Computer Society of India.