Agile Development using Ruby on Rails Framework

DARSHANA VIKAS WAGHMARE[1], PROF. PRATIBHA ADKAR[2]
[1,2] *MCA Department, PES Modern College of Engineering Pune, India*

Abstract -- Since Ruby on rails was established in 2004, ruby on rails is speedily becoming a powerful and standard framework for building web application. Rails is a development tool which gives web developers framework, providing structure for all the code they write. The Rails framework helps developers to build websites and applications, because it abstracts and simplifies common repetitive tasks. One of key principles of Ruby on Rails development is convention over configuration. This means that the programmer does not have to spend a lot of time configuring files in order to get setup, Ruby on Rails comes with a set of conventions which help speed up development. In this research paper we included detail information about Ruby on Rails like advantages of Ruby on Rails, MVC Architecture of Rails, Rails Framework and component of Ruby on Rails. Then explained Agile in detail. In agile we explained Agile Approach in Ruby on Rails, Agile Web Development Process, and explained Agile Development Methodology in Ruby on Rails for developing application. Then researcher has explained the web application which is develop in Ruby on Rails this application is develop by researcher.

Indexed Terms: Agile, Ruby on rails, Rails design architecture, MVC architecture, Agile Development

I. INTRODUCTION

Ruby on Rails is rapidly becoming one of the most powerful and popular frameworks for designing dynamic web application. Ruby on rails is open source web-based application framework composed using ruby programing language. It focusses on Model-Controller-View architecture. Ruby on Rails makes it simpler to assemble a database backend sustaining web applications that uses the language ruby.

The web application development is more tedious for software developers, to make it less tedious and more flexible the web development specialist uses the Ruby on Rails technology, which allows them to utilize the principles of agile methodology.

A. Ruby (Programing Language)

Ruby is pure object-oriented programming language. Which was developed in 1990 by Yukihiro Matsumoto (also known as Matz in Ruby community) in Japan. Ruby has similar syntax like C and Java, so it is easier for C and Java programmers to understand. It supports mostly all platforms like Windows, Linux and Mac. Ruby is based on many other languages like Perl, Lisp, Ads, Small talk and Eiffel. It is an interpreted scripting language which means most of its implementations execute instructions directly and freely, without previously compiling a program into machine-language instructions. The objective of Ruby's development was to make it act as a sensible buffer between human programmers and the underlying computing machinery.

B. Rails (Framework)

Rails is a server-side web application framework written in Ruby. Rails is a Model-Controller-View (MVC) framework, it provides default structure for database, a web service, and web pages. It encourages and facilitates web standards such as XML for data transfer, HTML, CSS, and JavaScript for user interfacing. Rails are open source framework for creating web application.

C. Advantages of Ruby on Rails

- One of the most notable advantages of Ruby on Rails is a strong backing of web standards for every part of application, from the user interface to data transfer.
- Applications that are made with Ruby on Rails are expected to follow established software design paradigms, such as "coding by convention", "don't repeat yourself", "active record pattern".
- Most popular websites in the world use Ruby on Rails. This puts it in the top 3 most used web development tools. It means that quite a

lot of people all over the world use this framework for creating software.

- Great number of helpful tools and libraries used in Ruby on Rails.
- Most of the time big companies with professional analysts make good decisions about the tools they use for their projects. This is why such tools often get an indisputable reputation of being solid and reliable.

II. RAILS DESIGN ARCHITECTURE

Rails focuses on programming pattern called MVC, which remains for Model-View-Controller.

A. MVC Architecture

Model View Controller is main feature of Ruby on Rails it is totally depends on it. 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 (Don't Repeat Yourself).

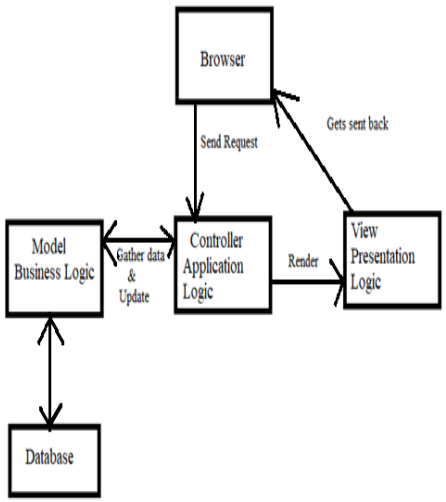

Figure. MVC Architecture

Model:

Maintains the relationship between Object and Database and also handle validation, association and transaction. It means that the model will maintain an extreme relation with the *Database*. Each model represent a database table. This model object gains capabilities to retrieve, save, edit, and delete data from database table. We use model objects as a layer between our application and the database. Besides that relation with the database, the *model* can create *validations* and *associations* between models.

View:

A presentation of data in a particular format, triggered by a controller's decision to present the data. It is the presentation of the request's response. This presentation can be a bunch of format types: PDF, HTML, JSON, etc. The final result of a view will be probably the user interface (UI)—Part of the "Client". For most pages on the web, the views will be an HTML styled with CSS and JS.

Controller:

The facility within the application that directs traffic, on the one hand querying the models for specific data, and on the other hand organizing that data (searching, sorting) into a form that fits the needs of a given view. The controller is the *"*Maestro*"*. It takes care of the flow: uses models to do queries, parses data, make decisions in which format you'll present the data.

B. Components of Rails

Ruby on Rails is a Model View Controller framework, with a pattern that's among the most useful when developing Web applications. The framework's main components are ActionPack, ActiveSupport, ActiveModel, ActiveRecord, ActiveResource, and ActionMailer. Essentially, Rails is a framework composed of other frameworks than can be used independently. Some details:

- Action Pack: Handles requests and responses. This framework is part of Rails's Model View Controller pattern, which serves the Web request, handling, routing, and view generation. To provide the response, it defines controllers that implement actions in order to render views.

- Active Model: Provide the interfaces for the Model part of Model View Controller. This

component is new in Rails 3.0. In previous versions, the model layer was based on Active Record. In the current version, you can use any ruby class as a Model. This is very important because you can use your own persistence layer and glue it into Rails.

- Active Record: This is the Object Relational Mapping (ORM) component of Rails, with a very nice zero-configuration feature. Naming and convention is the key to maintaining simple and minimal code to define classes that will be persisted in the database tables.

- Active Support: a collection of utility classes and standard library of extensions that are useful in Rails. We can find this extension useful for several Ruby projects.

- Active Resource: Connects business objects and Representational State Transfer (REST) Web services. With ActiveResource, you can easily use REST to expose your ActiveRecord models with just a small amount of code. This is a useful way to create an API without much effort.

- Action Mailer: This framework provides the email service layer, helping out to send the forgot password emails, registration emails, invoices for billing, etc. This class wraps ActionController from ActionPack to render the emails as page views, with the same render and templates like pages.

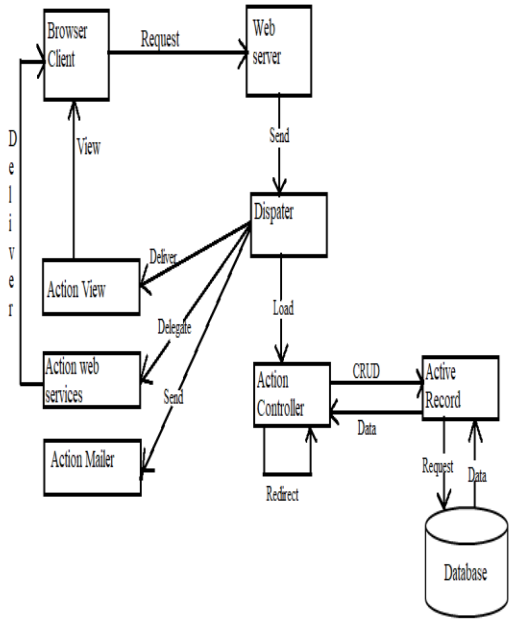

Figure. Architectural Model for Ruby on Rails (RoR)

III. AGILE

Agile is a term used to describe approaches to software development emphasizing incremental delivery, team collaboration, continual planning, and continual learning. Agile is not just a concept is a mind-set. A mind-set that drives an approach to software development. Agile methods (often called frameworks) are comprehensive approaches to phases of the software development lifecycle – planning, execution, and delivery. Agile is all about producing tangible, working results after each iteration.

A. Agile Approach in Ruby on Rails

Agile Manifesto as a set four preferences as given below: -

a) Individual and interaction over processes and tools:- Rails is all about individuals and interactions. Rails contains simple tools sets, no complex configurations, and no elaborate processes. There are small groups of developers, their favourite editor and chunks of Ruby code which leads to transparency and what the developers do is reflected immediately in what the customer sees. It is basically interactive process.

b) Working software over comprehensive documentation: - Rails does not support documentation and specification. Rails delivers working software early in the development cycle.

c) Customer collaboration over contract negotiation: - Rails project can quickly respond to changes required by the customer, who are thereby convinced that the team can deliver what is required, not just what has been requested.

d) Responding to Change Over Following a Plan: The Ruby on Rails is based on agile development, because of these reasons it can responding to the customer changes over following plan. The traditional method is not allowing to make a change in project plan. If developer make a mistake for doing a project plan and this mistake is occurred at the mid of project completion, then Rails is making a changes in whole project plan and correct the mistake by using agile.

B. Agile Web Development Process

The Agile web development process life-cycle is an iterative and incremental process life-cycle. Agile methods break major tasks into smaller number of subtasks called increments with minimal planning, and don't specifically involve long-term planning. 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.

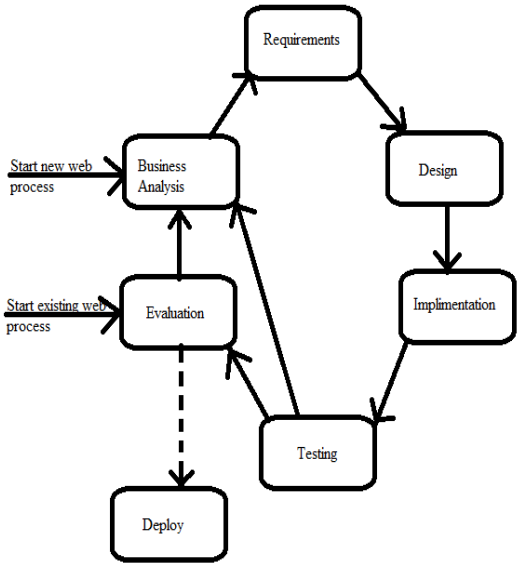

Figure. Agile Web Development Process

C. Agile Development Methodology for Building Ruby on Rails Application

1] List of Goals, Roles and Feature:

It will decided goal of whole project and also decided important features of project. It is decided which type of goal is achieve in feature.It also decides which person is doing which role in project. Each person have different views of same information of project. Also decide which is going on in site. Which type of features and categories is going to be add in this site is decides like User Registration using forums and blogging, Admin moderating the user content.

2] List of Stories:

A story is different than a feature because it represents a single thread of interaction from a particular user's perspective. Write these stories down on notecards. This will help you in estimation and prioritization.

3] Estimation of Story:

Estimation is a huge topic in itself, but the basic idea is to associate a particular level of effort with each story. Your goal here is to differentiate things that are low in effort, like stories that will result in you creating a simple model with a REST controller, from stories

that are high in effort, like interfacing your application with a challenging third-party API, or a story that will require you to use a technology you aren't very familiar with.

4] Prioritize the Stories:

Only the product owner can really make this decision. There are a lot of things that go into prioritization – deadlines, user testing, business value, etc. Estimation may have a lot to do with prioritization, because it illuminates opportunity cost.

5] Test the first story of completion:

Write Cucumber feature that covers the user's interaction with the site from beginning to end. Define the undefined steps as you come to them, and when you hit your first failure, you know that there's a behaviour that you desire that your app doesn't have.

6] Accept the Story:

Is the story acceptable? Does it do what you wanted it to? If not, you need to go back and make it work the way it was supposed to. Writing Cucumber tests in advance helps prevent this from happening.

7] Repeat until do:

This is how I do things. It's no means the only way to do things, but it is a very common way to do things in Rails. I think there's a good debate to be had around the value of agile estimation, or of particular technologies like Cucumber vs. Steak or RSpec vs Test::Unit, but most Rails developers will agree that the proper workflow is to identify a single story, write tests for it, and complete it.

IV. PROJECT DEVELOPED BY USING RUBY ON RAILS

Following project is developed by researcher using Ruby on Rails

Online Furniture Shopping is the process where consumer directly buy goods from a seller in real time, without an intermediary service, over the internet. It is a form of electronic commerce. This web application showcases all furniture products for shopping. To buy product the customer have an account, those who do not have an account make registration and create account, and then customer can login using id and password and also use this login id and password next time. Once the customer login then he will view all product and choose the product as per choice, customer add the product in cart for buy this product. Once customer add product into the cart the he can place the order, at the time of placing order he can write contact details, address, pin code and quantity of this product. After placing order view the payment option like card payment or cash on delivery. The customer choose the payment option as per his convenience.

CONCLUSION

Ruby on Rails is an open source web application development framework focused on Ruby programming languages and used extensively by agile programmer team has been popular for rapid web application development. The interesting aspect of RoR is programmer specify unusual configuration and as well as information used is unique and not duplicate. Rails is a perfect platform for Agile development practices. Ruby on Rails is extraordinary framework for creating web applications writing very little code in comparison.

REFERENCES

[1] S Meenakshi, "Ruby on Rails- An Agile Developer's Framework",Vol 112-No.1,February.
[2] Anmol kiran uttarkar, "Agile Developer's Framework: Ruby on Rails",Vol 8 ,December.
[3] Ruby S., Thomas D., Heinemeier Hansson D. Agile Web Development with Rails, 4th Edition, Rails 3.2 (The Pragmatic Programmers) – 2011
[4] ANDREW MCDONALD, "AGILE WEB ENGINEERING (AWE) PROCESS: PERCEPTIONS WITHIN AFORTUNE 500 FINANCIAL SERVICES COMPANY", Joural of Web Engineering Vol 4, No. 4, 2005.
[5] Andrew Mcdonald,"AgileWeb Engineering(AWE)Process: Perceptions

Within A Fortune 500FinancialServicesCompany"JournalOf WebEngineering,Vol.4,No.4 (2005) 283-312.

[6] Martin Fowler "New Methodology", http://martinfowler.com/articles/ newMethodology.htmlVol.1de17(19/08/2010).

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

[8] Aspire Systems, "Ruby on Rails – Building Dynamic Web Products Faster", 2011.

[9] Shubham Devgire, Rohan Jagtap, "Review on Ruby on Rails Developers–An Agile Developer'sFramework", in journal of Web Development and Web Designing, Volume 1 No2, 2016

[10] Malik Hneif, Siew Hock Ow, "Review Of Agile Methodologies In Software Development", International Journal of Research and Reviews in Applied Sciences ISSN: 2076-734 X, EISSN : 2076-7366 Volume1, Issue 1(October 2009).

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

[12] 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.

[13] 13] Phalnikar, Deshpand, &Joshi, Applying Agile Principles for Distributed Software Development – International Conferenceon Advancesin Computingand Communications 01/2009; DOI:10.1109/ICACC.2009.93 ISBN:978-0-7695-3516-6.

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

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