Ruby on Rails, причины успеха – взгляд изнутри

Васильева Татьяна

Ланит-Терком

Web-приложения

- Доступность
- Простота установки и поддержки
- Новые технологии, позволяющие реализовывать в Web сложную логику

Ruby on Rails (RoR)

- Программный каркас (framework), позволяющий легко создавать, разворачивать и поддерживать web-приложения
- Открытое программное обеспечение
- Стремительно развивающая технология, ставшая очень популярной

Причины успеха

- Ruby on Rails – это просто
- Ruby on Rails – это Agile
- Ruby on Rails – это Web 2.0

Причины успеха

- **Ruby on Rails – это просто**
- Ruby on Rails – это Agile
- Ruby on Rails – это Web 2.0

Язык Ruby

```
class Lesson < ActiveRecord::Base
   belongs_to :course
   has_one :teacher
   has_and_belongs_to_many :learners

   validates_presence_of :name,:description
   validates_uniqueness_of :short_name
end
```

Основные принципы RoR

- MVC как предопределенная архитектура приложения
- Связывание компонент без использования конфигурационных файлов (принцип «Convention over Configuration»)
- Применение механизмов повторного использования (принцип «Don’t Repeat Yourself»)
- Простота за счет гибкости

RoR и базы данных [1]

- Генерация каркаса приложения по существующей схеме БД
- Возможность внесения изменений в схему БД средствами RoR
- Версионирование схемы БД вместе с кодом приложения
- Отсутствие интеграции с каким-либо CASE-средством

[1] В качестве сервера баз данных поддерживаются MySQL, PostgreSQL, SQLite, IBM DB2, Oracle и Microsoft SQL

Причины успеха

- Ruby on Rails – это просто
- **Ruby on Rails – это Agile**
- Ruby on Rails – это Web 2.0

Agile

- Часто меняющиеся требования
- Необходимость быстрого реагирования на изменения
- Работающее ПО как основная мера развития проекта
- Минимальное количество документации
- Тесное общение с заказчиком
- Эффективность за счет предсказуемости

Ruby on Rails – это Agile (1)

Часто меняющиеся требования и необходимость быстрого реагирования на изменения

- Каркас приложения и MVC позволяют легко выделить фрагменты кода, которые необходимо изменить
- Простота RoR позволяет быстро внести изменения
- Общепринятые соглашения позволяют легко вводить новых людей

Ruby on Rails – это Agile (2)

Работающее ПО как основная мера развития проекта

- Генерация каркаса приложения по БД позволяет получить работающее ПО на ранних стадиях разработки
- Готовый каркас приложения позволяет усложнять и наращивать систему небольшими итерациями, результатом каждой из которых является работающее приложение
- RoR предоставляет средства тестирования приложения с первых дней разработки

Ruby on Rails – это Agile (3)

Минимальное количество документации

- RoR позволяет генерировать документацию приложения

Тесное общение с заказчиком

- Наличие специального web-сервера WEBrick дает возможность быстро развернуть приложение и показать нечто работающее заказчику уже на самых ранних этапах разработки
- Возможность быстро вносить изменения зачастую переводит общение с заказчиком в форму «Что, если?»

Причины успеха

- Ruby on Rails – это просто
- Ruby on Rails – это Agile
- **Ruby on Rails – это Web 2.0**

Ruby on Rails – это Web 2.0 (1)

- AJAX(Asynchronous JavaScript and XML) – подход к построению пользовательских интерфейсов web-приложений
- RoR предоставляет средства поддержки AJAX, которые позволяют создавать AJAX объекты в представлениях
- Средства поддержки достаточно эффективны и удобны при решении целого ряда задач (постраничный вывод, быстрый поиск)

Ruby on Rails – это Web 2.0 (2)

- Но:
  - Создавая форму, необходимо помнить о том, что обновлять можно лишь элемент
  - Для решения некоторых задач приходится использовать «чистый» AJAX, что противоречит принципу DRY

Заключение

- Не является «серебряной пулей»
- Вопрос эффективности – вопрос применимости в каждом конкретном случае
- Выбор наиболее оправдан при разработке:
  - Web 2.0 приложений
  - Приложений, представляющих собой интерфейс к базе данных со сложной логикой
  - Приложений, где важны быстрое реагирование на изменения и эффективность команды разработчиков