Finding the Right Tool For the Job:
Agile Development for Ruby on Rails

Presented by: Faith-Anne Kocadag • fk4687@stu.armstrong.edu

I. Abstract:

This project outlines the significant components involved in choosing a relational database design. It compares two popular web development paradigms and employs the Agile Software Development model for project management in the implementation the MOSH database.

Agile Software Development is an iterative software project management model chosen for its flexibility and rapid prototyping. The adaptive incremental iterations of this model allow for both client and developer evaluations between cycles.

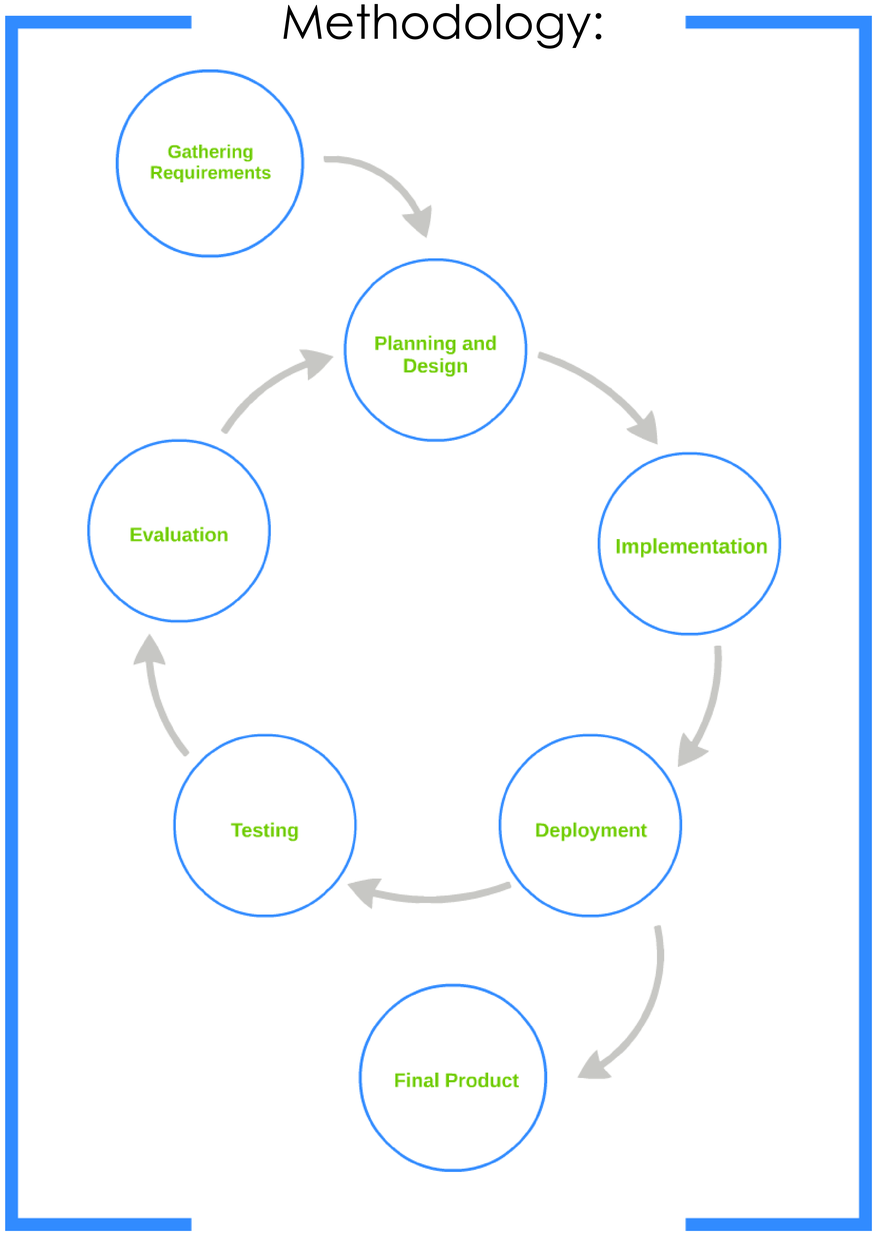
Ruby on Rails is an increasingly popular framework for web applications that interface with a database. Its focus on convention over configuration lends itself to rapid prototyping.

MOSH, the McCarthy Online Support Hub, is a Rails application created for the faculty and staff of the Department of Computer Science and Information Technology at AASU. It automates common department tasks like inventorying hardware, handling software imaging, managing work orders and poster orders, and keeping track of an after hours lab access list. These jobs were previously handled manually through paper and email chains.

II. Tasks:

1. Compare Ruby on Rails to another popular web framework, LAMP, in terms of architecture, ease of implementation, and security.
2. Use Agile Development processes to develop a web database.

Methodology:



III. Aim:

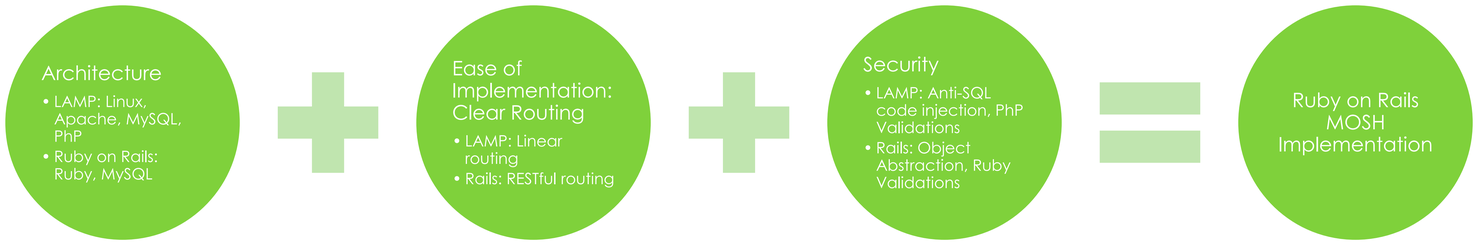
To develop a web database for use by the AASU Department of Computer Science and Information Technology faculty and staff.

IV. Results:

Ruby on Rails was chosen over other frameworks, like LAMP, because of its:

- *Ease of implementation: clear routing system*
  - Rails uses RESTful routing for the most common web database tasks. Users can directly access the pages with the services they require without navigating through complicated routes. RESTful routes include:
    - *tablename.new.html
    - *tablename.edit.html
    - *tablename.show.html
    - *tablename.index.html
  - LAMP has no conventions for routing. It takes more planning to design an application that is intuitive and flexible enough to allow for nonlinear site flow.
- *Enhanced Security*
  - Rails' Model View Controller framework uses layer separation and object abstraction to prevent potentially harmful direct access to the database by unauthorized users.
  - LAMP has no seamless framework for layer separation. Type specific code to prevent SQL injections and other security issues must be manually inserted to prevent corresponding security issues.

Frameworks: Rails over LAMP



V. Conclusions:

Choosing the proper tools for a job is crucial from maximum efficiency and usability. Ruby on Rails proved to be the most apt framework for the creation of the MOSH database application. It was chosen because of its use of abstraction and highly structured model – view – controller architecture, and its highly intuitive routing systems. This focus on convention over configuration lent itself well toward the Agile project management model, and allowed for rapid prototyping and increased client feedback. Increased feedback ensures that the finished product most closely matches client expectations.