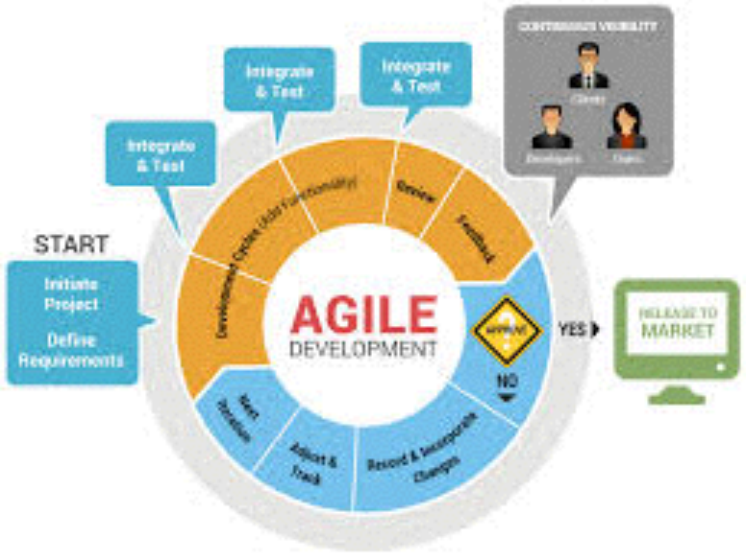
AUT UNIVERSITY COMPUTER + MATHEMATICAL SCIENCES



Week 3: Intro to Agile SDM and Rails

409232 Software Development Methods

Jim Buchan

SERL

Software Engineering Research Laboratory

Quizz

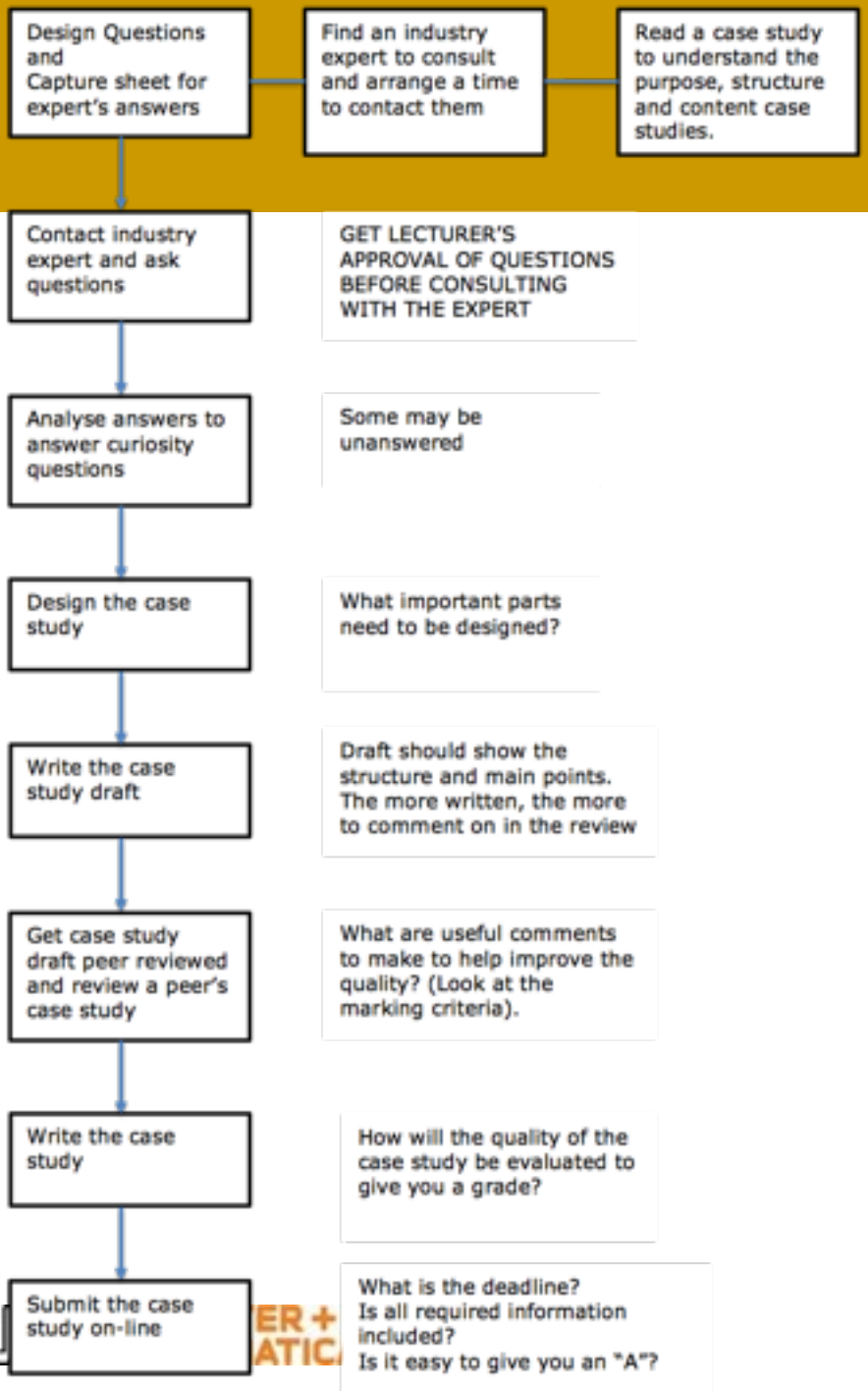
Today's Goals

- Clarity on Assignments 1 and 2
- Overview of Rails and Ruby

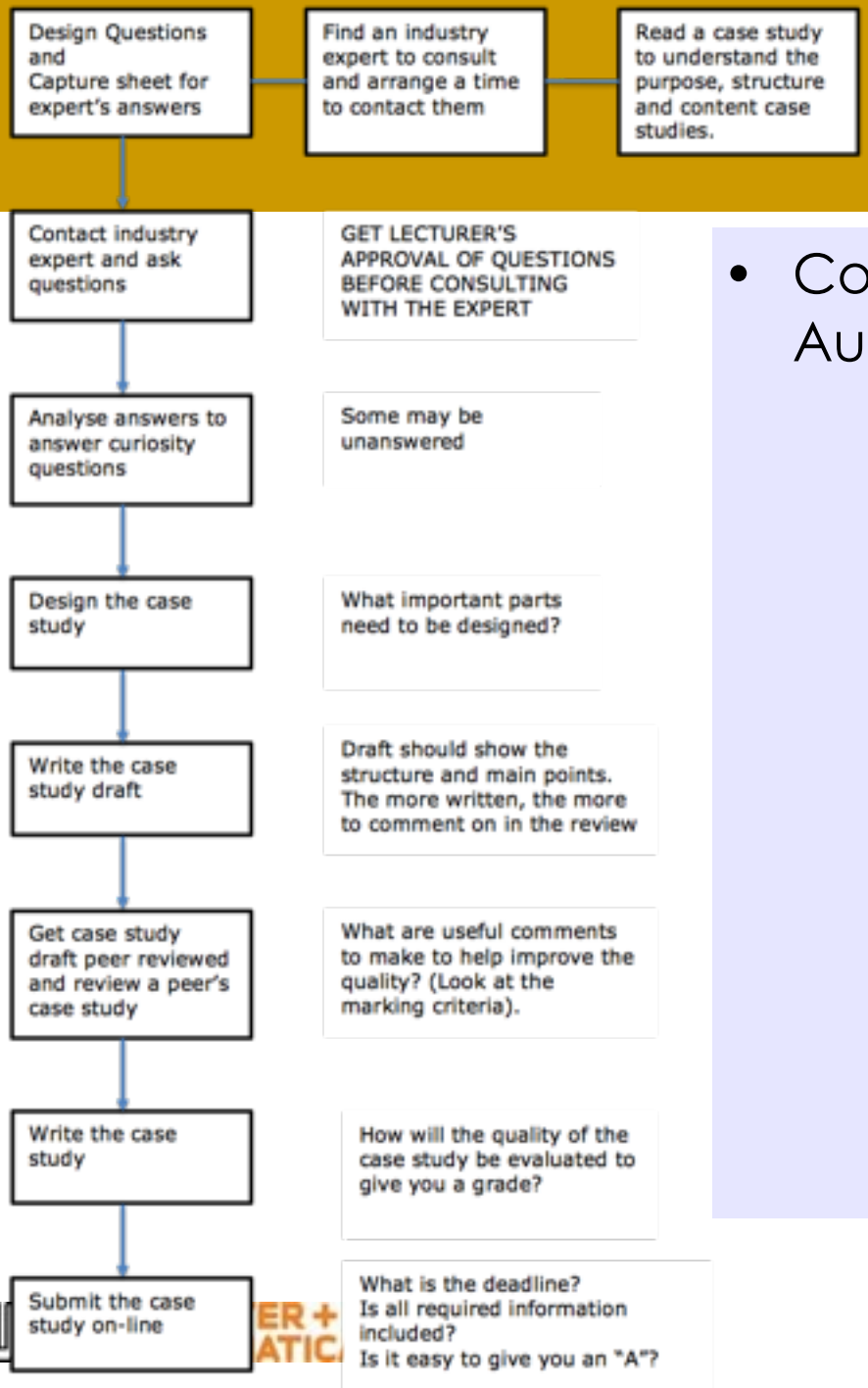
Assignment 1

- You will write a Fictional Case Study based on insights from consulting Software Development expert from industry
- You should consult with an industry expert to get their opinion on what the important software development methods, practices and tools you should learn about.
- You can use your own personal contacts or contact your lecturer who may also have some software developer contacts in industry willing to be consulted.

Assignment 1

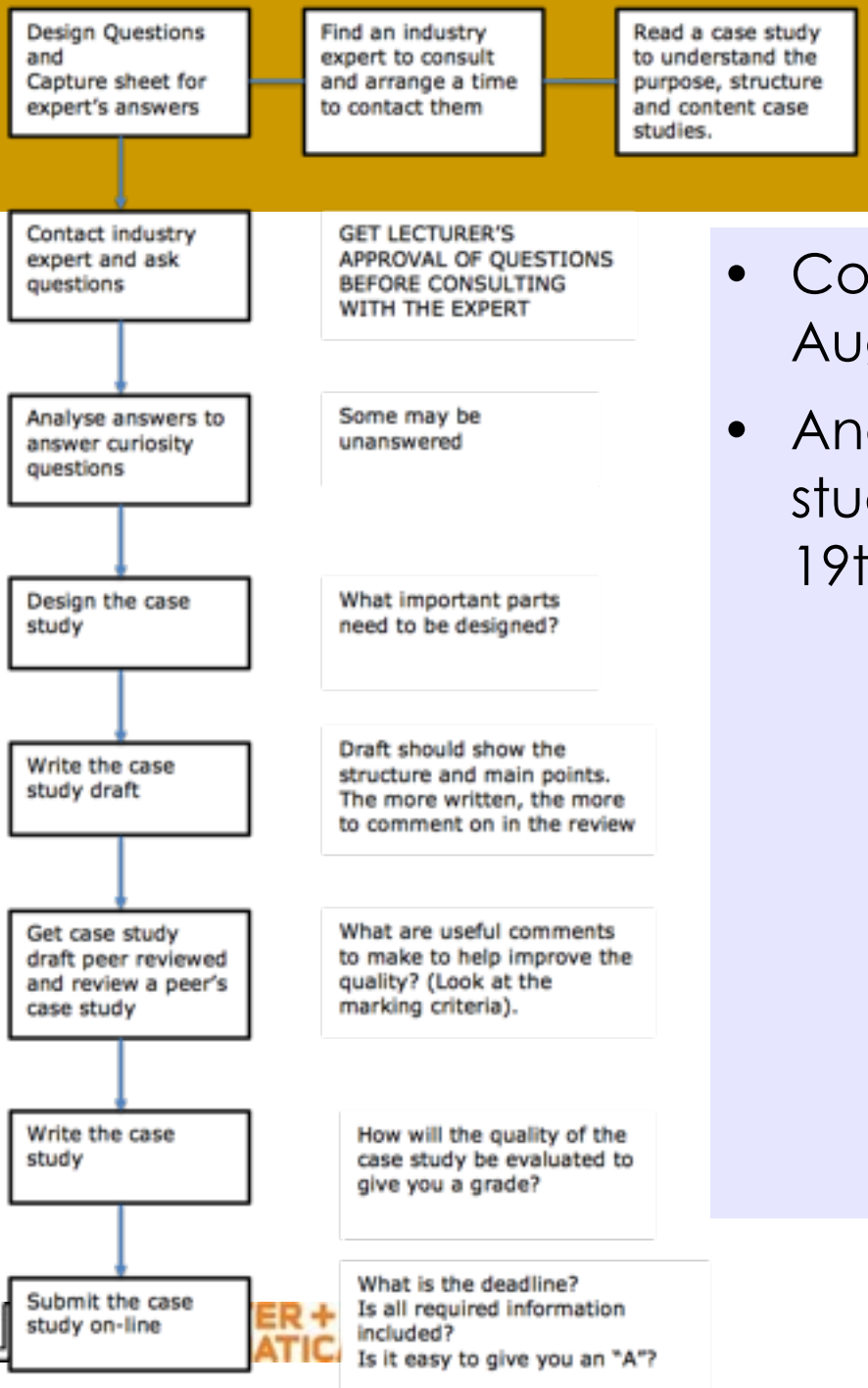


Assignment 1



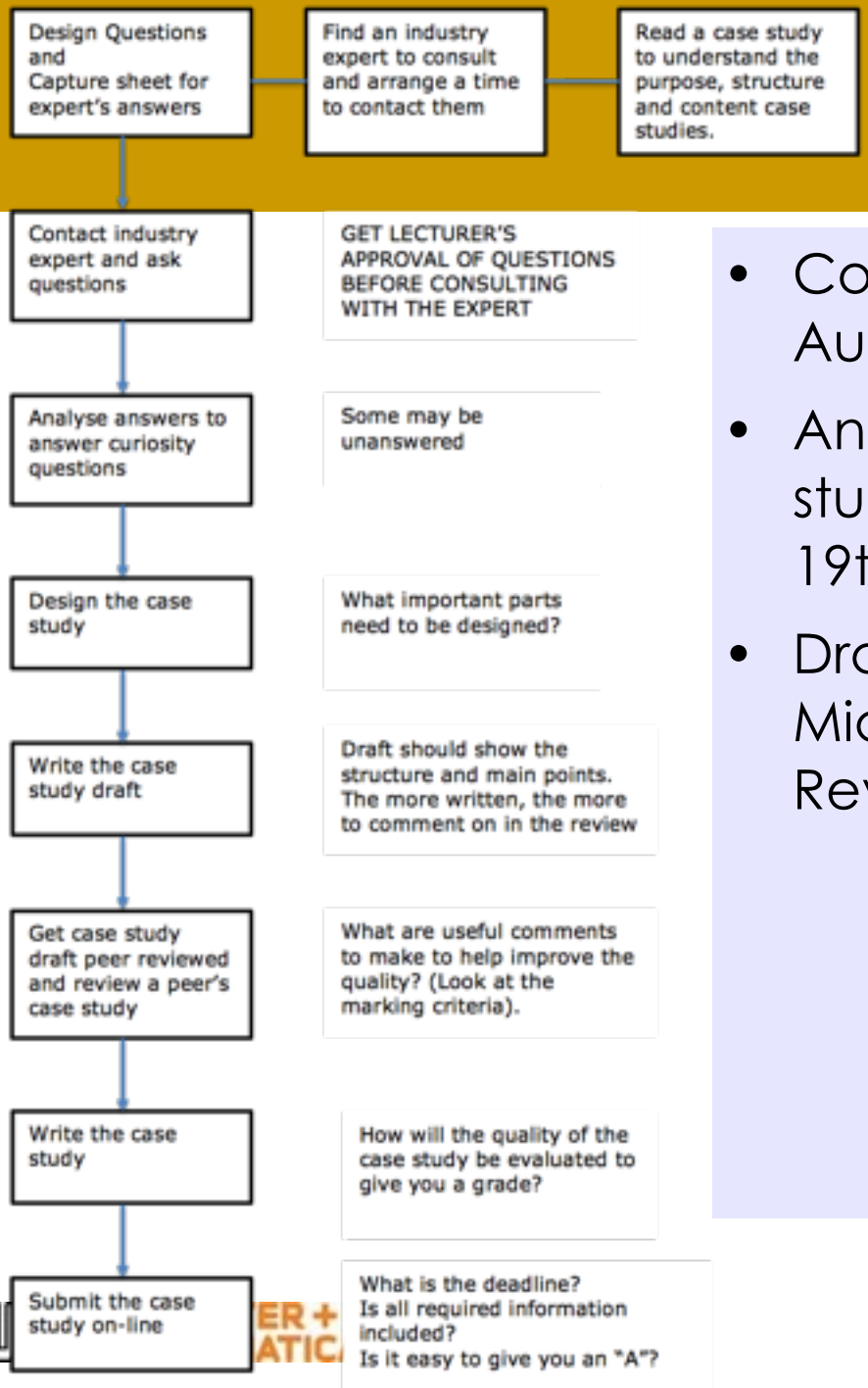
- Consultations finished by Friday August 7th

Assignment 1



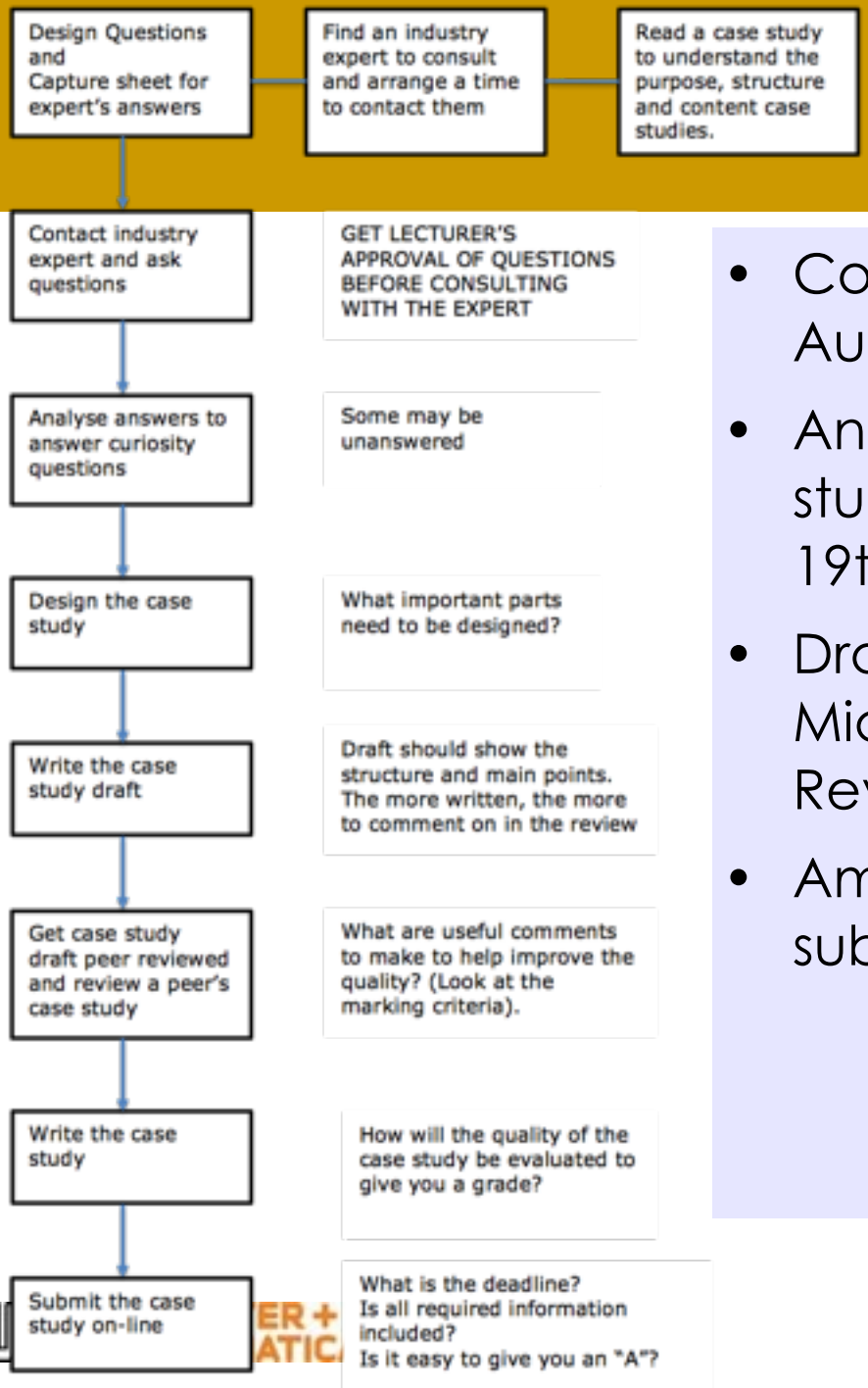
- Consultations finished by Friday August 7th
- Analysis of answers and draft case study written by Wednesday August 19th

Assignment 1



- Consultations finished by Friday August 7th
- Analysis of answers and draft case study written by Wednesday August 19th
- Draft available for peer review Midday Wednesday August 19th. Reviewed by Friday August 21st.

Assignment 1



- Consultations finished by Friday August 7th
- Analysis of answers and draft case study written by Wednesday August 19th
- Draft available for peer review Midday Wednesday August 19th. Reviewed by Friday August 21st.
- Amendments and final case submitted by Friday August 28th

SERL
Software Engineering Research Laboratory

Assignment 2

- Create an application for which I am the Product Owner
- SaaS over web
- Requirements.
  - User stories
  - Cucumber scenarios.
- Version control and continuous integration
- testing - auto unit test, auto build
- use Rails

Assignment 2

- Sprint 0 - weeks 3,4,5
- Sprint 1 weeks 6,7
- Sprint 2 weeks 8,9
- Sprint 3 weeks 10,11
- Delivery week 12
- 6 Teams of 6-8 members
  - Analyst x 1, testers x 2, coders x 2, scrum master x 1
  - 2 competing streams of 3 teams developing different versions of the app.
    - 1 team - input of information
    - 1 team - querying information
    - 1 team - workflow and administration

In Sprint 0 - set up

- Use of Rails, BDD
- Use of RSpec for automated testing
- Use of Travis CI for continuous integration
- Use of Cucumber for acceptance testing
- Use of Bootstrap, Font awesome for look and feel
- Use of GitHub for version control and code sharing
- Use of Heroku for cloud deployment
- Get product backlog - manage in Trello
- Sprint planning, release planning

Agile organisational Methods

- Teams/individuals share code within each stream
- Standup meeting every class
- Sprint planning before each sprint (priority and estimation)
- Pair programming once a week
- Showcase at the end of each sprint
- Teams keep in touch f2f and electronically on a DAILY basis

Agile organisational Methods

- Teams/individuals share code within each stream
- Standup meeting every class
- Sprint planning before eac (priority and estimation)
- Pair pr ing once a week
- release at the end of each sprint
- Teams keep in touch f2f and electronically on a DAILY basis

How evidence all of this for academic credit?

Agile organisational Methods

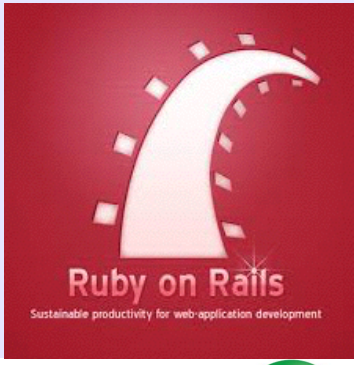
- Teams/individuals share code within each stream
- Standup meeting every class
- Sprint planning before eac
and estimation)
- Pair pr
week
end of each sprint
- Te keep in touch f2f and electronically on a DAILY basis

How evidence all of this for academic credit?

How document individual contribution?

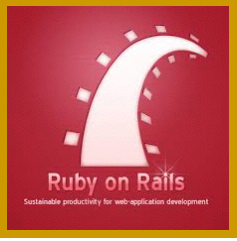
Ruby and Rails

- Ruby is a general purpose programming language, widely used for web programming
  - 20 years old, developed by Yukhiro Matsumoto
  - Popular because of the software library Rails written in Ruby that extends the Ruby language



Ruby and Rails

- Rails (Ruby on Rails) is library of software packages(a RubyGem or “gem”) that provides a framework for simplifying the building of web sites.
  - Rails is used by using the Rails API
  - Rails combines Ruby with HTML, CSS and Javascript to create a web app that runs on a web server
  - Rails is a “back-end” or server-side web app development framework
  - Rails has conventions and rules that make collaboration with other Rails developers easy and is well suited to Agile



Ruby and Rails

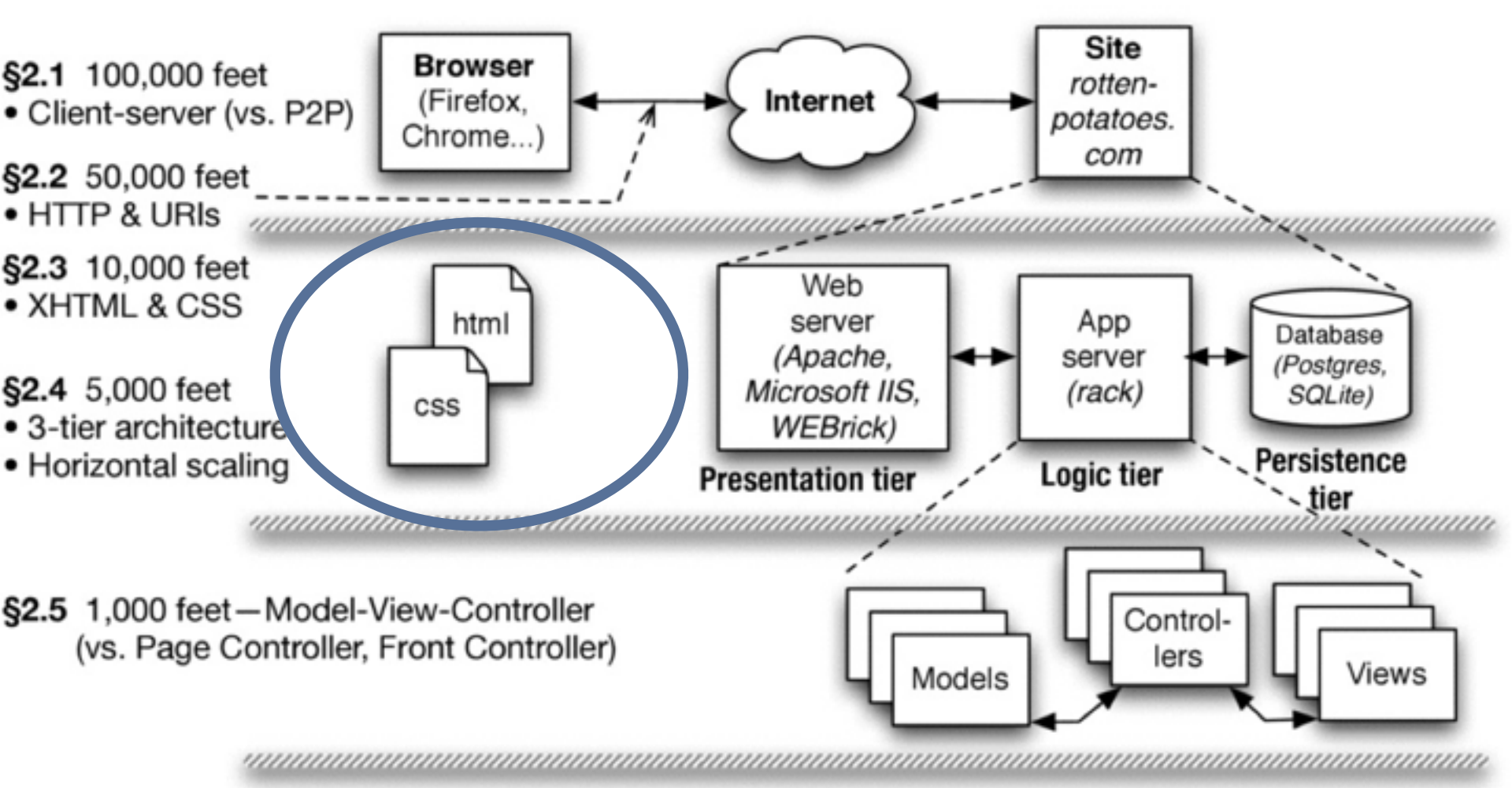
- Convention over configuration
  - e.g. in Rails create a model object “User”, it will save data to a database table named “users”
- Don’t Repeat Yourself (DRY)
- avoid duplication of code - Rails uses Ruby’s metaprogramming to re-use code

Gemfile

- Every application has a GemFile in the root folder that lists the each gem used by the application

Rails and testing

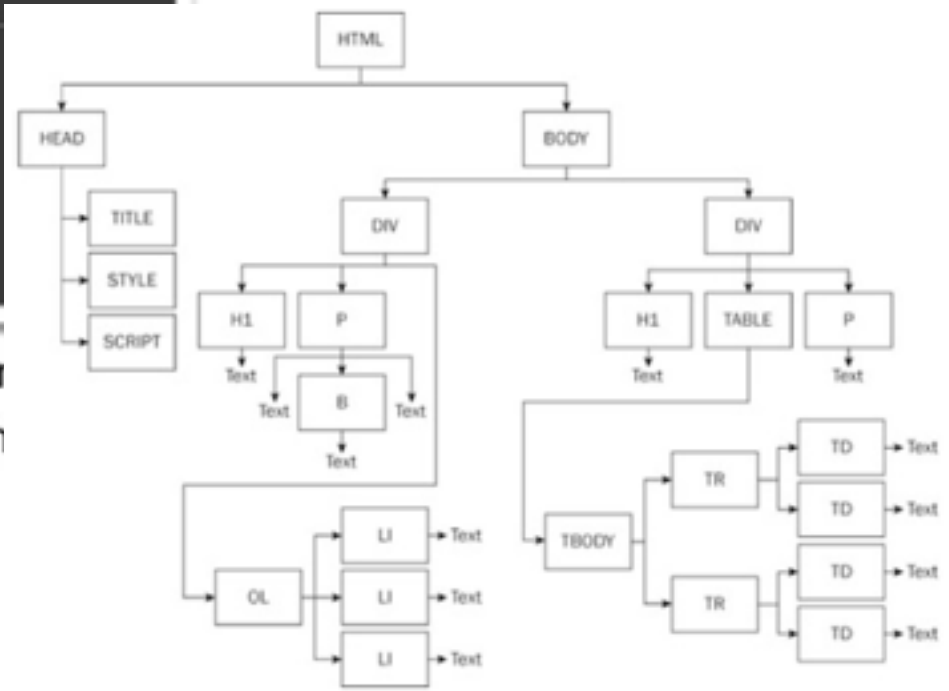
- Test Driven Development
  - Built in or common to use RSpec gem
  - Automated set of unit tests
- Behaviour Driven Development
  - write specifications in the form of descriptive stories that are the basis for automated acceptance tests
- Integration testing
  - test all components work (pass tests) when assembled together



§2.6 500 feet: Active Record models (vs. Data Mapper) • **Active Record** • **REST** • **Template View**
§2.7 500 feet: RESTful controllers (Representational State Transfer for self-contained actions) • Data Mapper • Transform View
§2.8 500 feet: Template View (vs. Transform View)

Hypertext Markup Language (HTML)

```
<p>This is an element</p>

<br /><!-- comment-->

<img src="welcome.jpg" id="welcome"/>

<h1 class=" foo" >
This is an element with an attribute
</h1>
```



- Document = Hierarchical collection of ele
- Element can have attributes (many option

Document Object Model (DOM)
The elements of a page are nested into a tree-like structure of objects

Cascading Style Sheets (CSS)

The visual appearance is described in a separate document call a style sheet

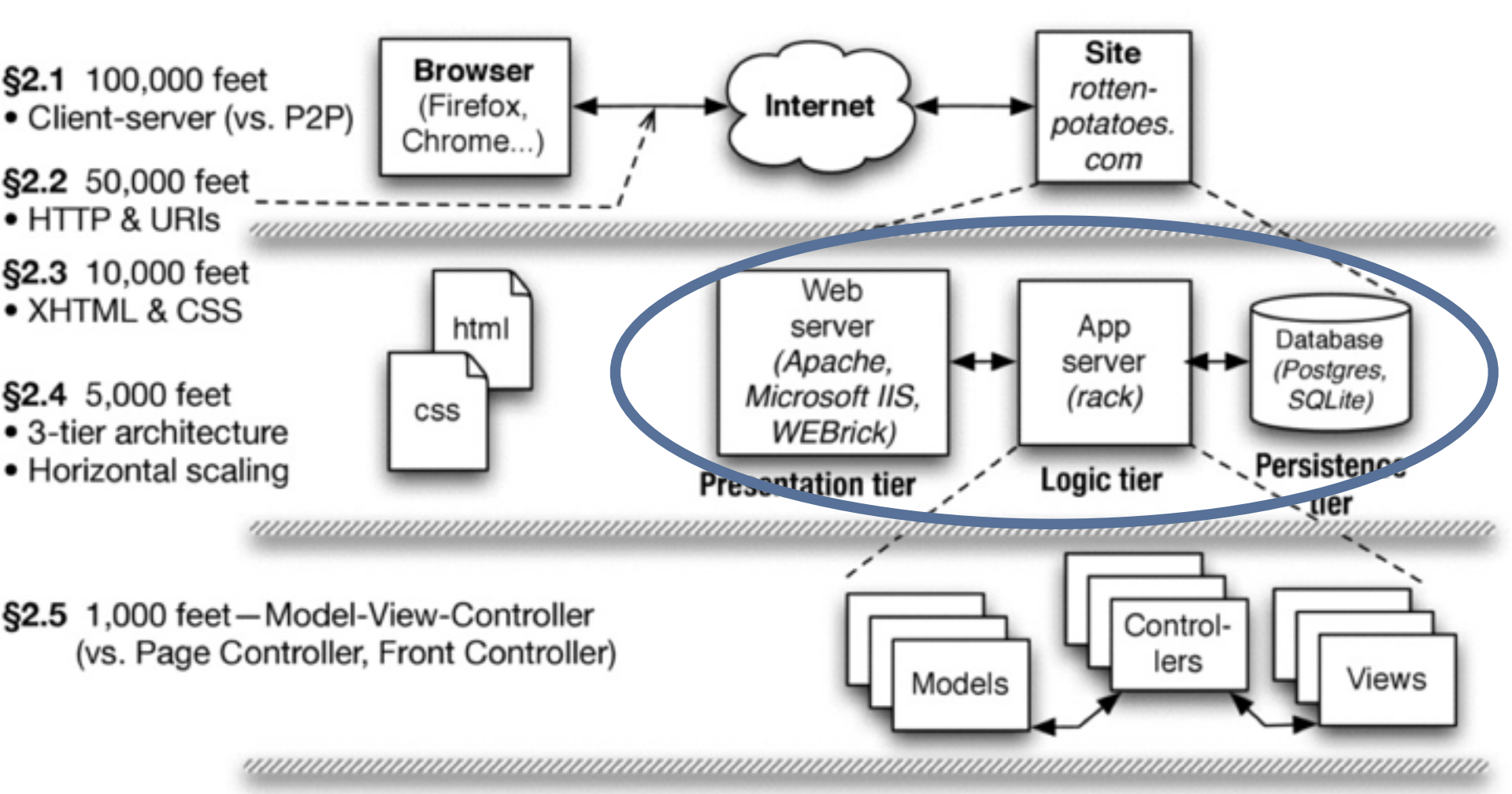
HTML should contain NO visual styling to separate

HTML id and class attributes are used in CSS files

```
// id selector
#main { background-color: orange;}

// class selector
.sidebar { color: black; }

// element selector
span { font-size: 24px;}

// mixed
span.sidebar { color: #C5C5C5; }
```

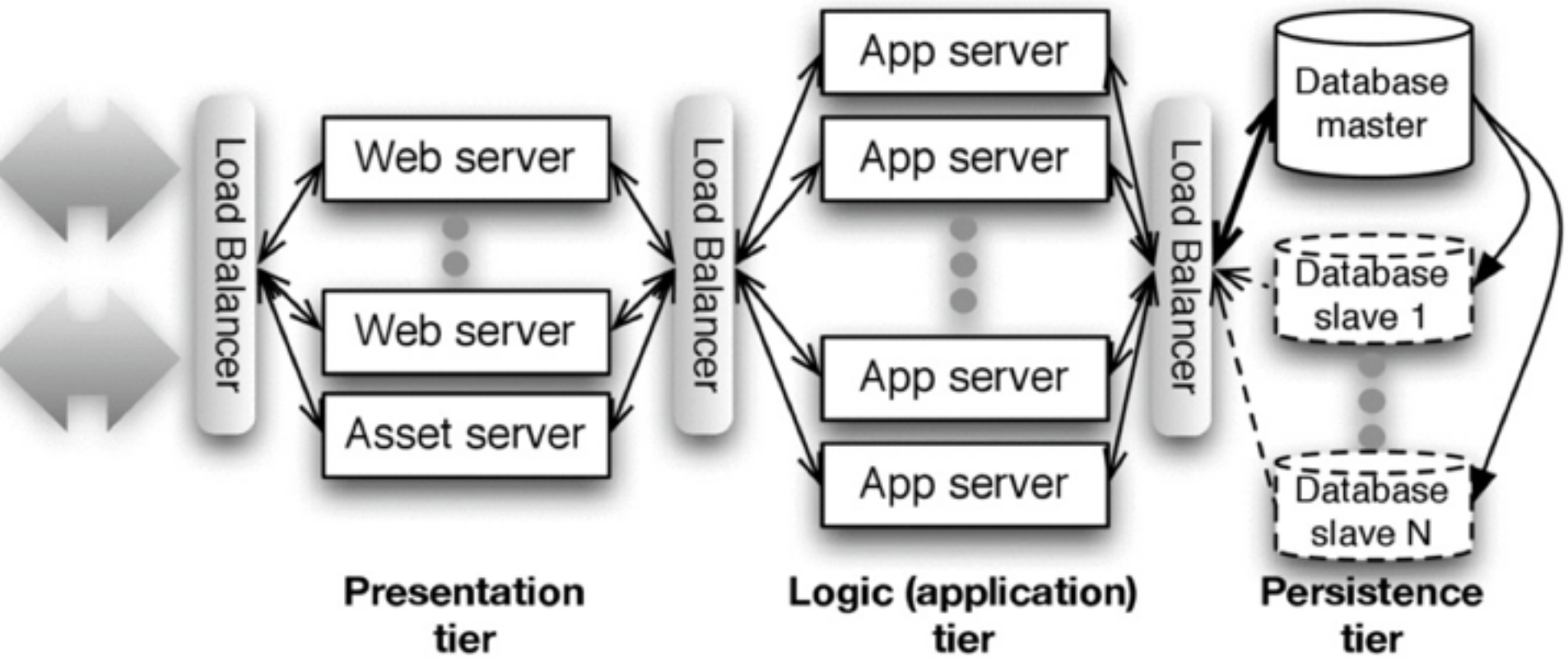
In a Rails application

- app/assets directory contains CSS files, Javascript files, images
- Can use gems such as Bootstrap and Font Awesome to simplify
  - http://fortawesome.github.io/Font-Awesome/
  - http://getbootstrap.com
- templating with ERB and HAML
  - http://haml.info/tutorial.html

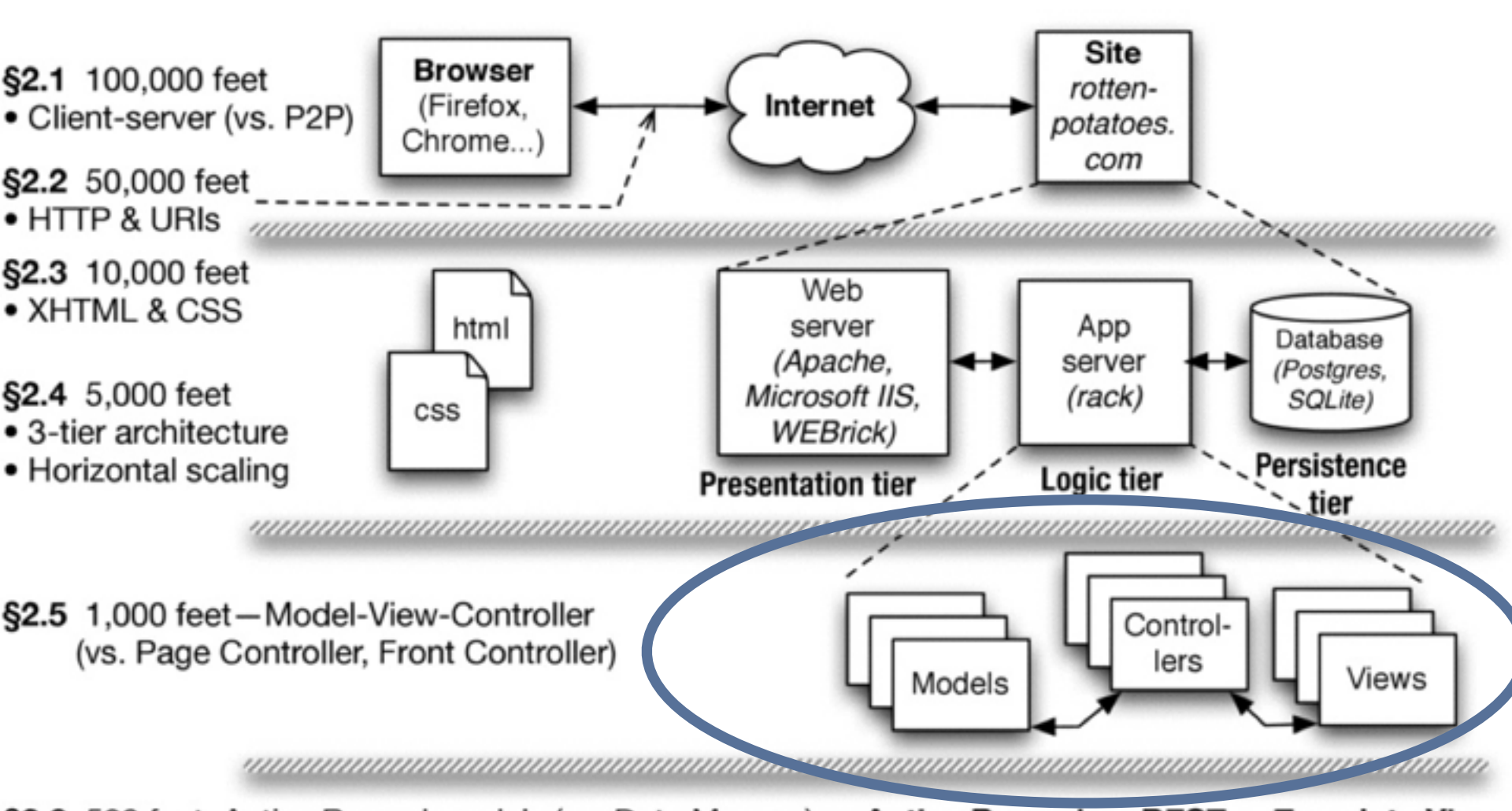


§2.6 500 feet: Active Record models (vs. Data Mapper)
§2.7 500 feet: RESTful controllers (Representational State Transfer for self-contained actions)
§2.8 500 feet: Template View (vs. Transform View)

- **Active Record**
- Data Mapper
- **REST**
- **Template View**
- Transform View



The 3-tier ***shared-nothing*** architecture allows adding computers at any tier independently to match demand (more complex at persistence layer)

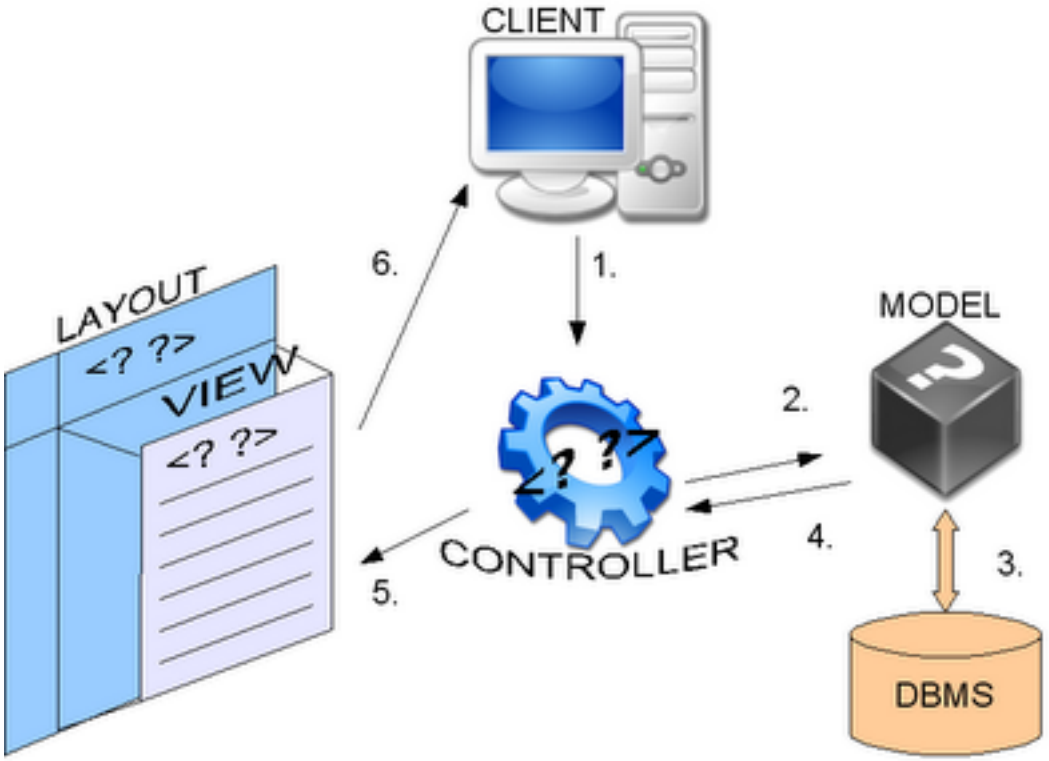


§2.6 500 feet: Active Record models (vs. Data Mapper)
§2.7 500 feet: RESTful controllers (Representational State Transfer for self-contained actions)
§2.8 500 feet: Template View (vs. Transform View)

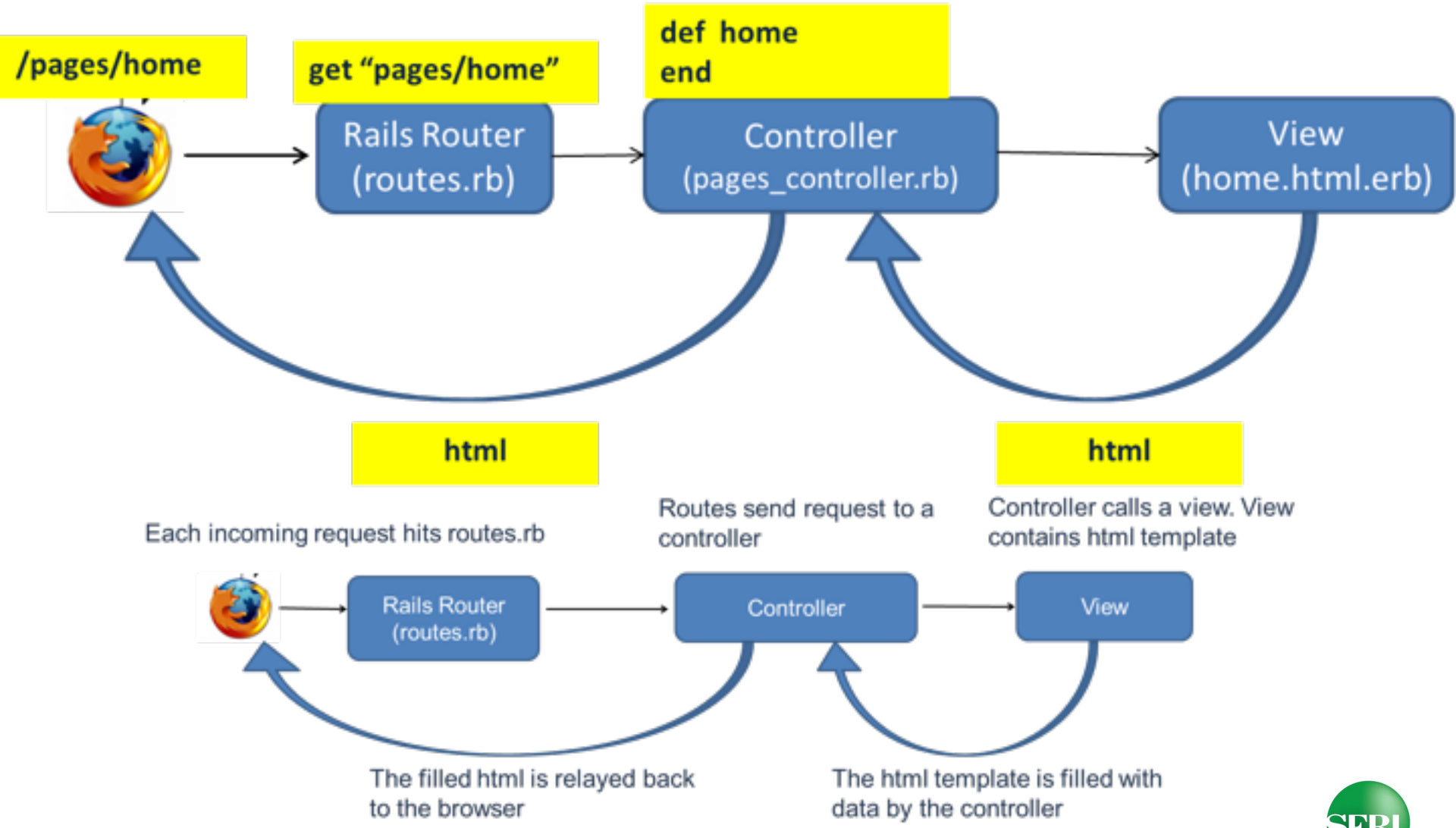
- **Active Record**
- Data Mapper
- **REST**
- **Template View**
- Transform View

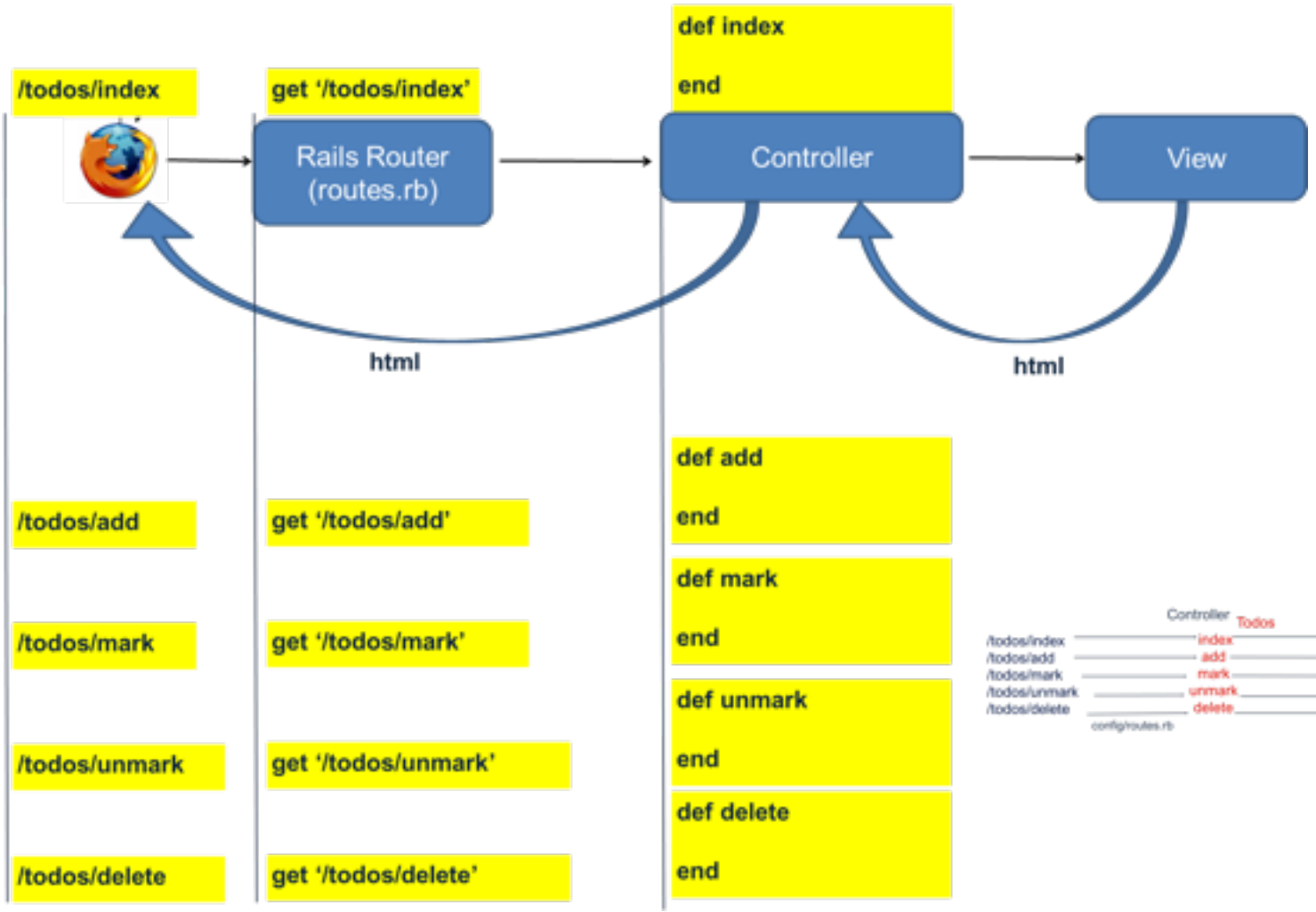
In a Rails application

- Isolation of business logic from the user interface
- Ease of keeping code DRY
- Making it clear where different types of code belong for easier maintenance



In a Rails application




/todos/index
get '/todos/index'
def index
end
Rails Router
(routes.rb)
Controller
View
html
html
/todos/add
get '/todos/add'
def add
end
/todos/mark
get '/todos/mark'
def mark
end
/todos/unmark
get '/todos/unmark'
def unmark
end
/todos/delete
get '/todos/delete'
def delete
end
Controller
Todos
/todos/index
index
/todos/add
add
/todos/mark
mark
/todos/unmark
unmark
/todos/delete
delete
config/routes.rb
Output HTML


Controller Todos
/todos/index
/todos/add
/todos/mark
/todos/unmark
/todos/delete
index
add
mark
unmark
delete
index.html.erb
add.html.erb
mark.html.erb
unmark.html.erb
delete.html.erb
Output HTML
(we have not gone into specifics of this yet)


Controller Todos
/todos/index
/todos/add
/todos/mark
/todos/unmark
/todos/delete
index
add
mark
unmark
delete
index.html.erb
List of todos on the browser
add.html.erb
mark.html.erb
unmark.html.erb
delete.html.erb

Rails Tutorials

- http://guides.rubyonrails.org/getting_started.html
- https://www.railstutorial.org/book/beginning
- Collect tutorials and other resources on Blackboard
- Use Discussion boards on Blackboard

Practical exercise 3

- This weeks practical exercise involves setting up Bootstrap and editing some .haml files to change the format of the web pages served up.
- Make sure you have added all the necessary files to github tracking (git add …) including the Gemfile. Then commit changes (git commit -am"exercise 3") and push to the cloud repository (git push).
- You should have a look at Awesome Font and see if you can improve the icons in rottenpotatoes using this resource.
- Then deploy your updated application to the cloud using Heroku and confirm it works.