



CO-PRESENTED BY  & O'REILLY

Refactoring Your Rails Application

RailsConf 2008

Zach Dennis

Mutually Human Software
www.mutuallyhuman.com

Drew Colthorp

Atomic Object
www.atomicobject.com



CO-PRESENTED BY  & O'REILLY

Example

Extract Query to Model



CO-PRESENTED BY  & O'REILLY

What is refactoring?



CO-PRESENTED BY  & O'REILLY

Why refactor?



CO-PRESENTED BY  & O'REILLY

Why refactor?

Program Evolution



CO-PRESENTED BY  & O'REILLY

Why refactor?

Technical Debt / Pain Management



CO-PRESENTED BY  & O'REILLY

How do you refactor?



CO-PRESENTED BY  & O'REILLY

How do you refactor?

Step #1 - have solid tests



CO-PRESENTED BY  & O'REILLY

How do you refactor?

Short deliberate steps



CO-PRESENTED BY  & O'REILLY

How do you refactor?

Wear two hats, but only one at a time.



CO-PRESENTED BY  & O'REILLY

Testing

Automated test suite preferred



CO-PRESENTED BY  & O'REILLY

Testing

When in doubt nothing beats a good manual test plan



CO-PRESENTED BY  & O'REILLY

What do you refactor?



CO-PRESENTED BY  & O'REILLY

What do you refactor?

Code that smells



CO-PRESENTED BY  & O'REILLY

Common smells in Rails apps

Duplicated Code

Shotgun Surgery

Long Method

Inappropriate Intimacy

Divergent Change

Large Class

Feature Envy

Message Chains

Comments



CO-PRESENTED BY  & O'REILLY

Duplicated Code smell

```
def organizer_names  
  Organizer.find(:all).map(&:name)  
end
```

```
def organizer_ids  
  Organizer.find(:all).map(&:id)  
end
```



CO-PRESENTED BY  & O'REILLY

Shotgun Surgery smell

A change requires modifying many different parts of your application.



CO-PRESENTED BY  & O'REILLY

Long Method smell

```
1: def create
2:   @record = Record.find(params[:id])
3:   @record.name = params[:name]
...
37: if @record.save
38:   redirect_to records_path
39: else
40:   render :action => "edit"
41: end
42:end
```



CO-PRESENTED BY  & O'REILLY

Inappropriate Intimacy smell

```
class PeopleController
  def index
    Person.find(:all,
      :select => "name",
      :include => [:companies, :groups]
      :order => :updated_at
    )
  end
end
```



CO-PRESENTED BY  & O'REILLY

Divergent Change smell

A class is commonly changing in different ways for different reasons.



CO-PRESENTED BY  & O'REILLY

Large Class smell

A class is doing too much.

Common indicators:

- too many instance variables**
- too much code**



CO-PRESENTED BY  & O'REILLY

Feature Envy smell

A method is more interested in a class other than the one it is actually on.



CO-PRESENTED BY  & O'REILLY

Many more smells exist

<http://www.refactoring.com/sources.html>



CO-PRESENTED BY  & O'REILLY

Deliberate code smells

Code smells have their tradeoffs



CO-PRESENTED BY  & O'REILLY

When do you refactor?



CO-PRESENTED BY  & O'REILLY

When do you refactor?

Adding a feature



CO-PRESENTED BY  & O'REILLY

When do you refactor?

Cleaning up the code base



CO-PRESENTED BY  & O'REILLY

When do you refactor?

Spikes, prototyping



CO-PRESENTED BY  & O'REILLY

Refactoring is essential

Red. Green. Refactor.



CO-PRESENTED BY  & O'REILLY

Common Rails Dilemma

MVC paralysis



CO-PRESENTED BY  & O'REILLY

M



V



C





CO-PRESENTED BY  & O'REILLY

M



V



C





CO-PRESENTED BY  & O'REILLY

Rails Refactorings



CO-PRESENTED BY  & O'REILLY

Example

Move Data From View Into Presenter



CO-PRESENTED BY  & O'REILLY

What is a Presenter?

An object responsible for view-related logic and behavior.



CO-PRESENTED BY  & O'REILLY

**Presenters are
decorators,
they wrap objects,
and delegate**



CO-PRESENTED BY  & O'REILLY

**Presenters can work for
individual resources /
models**

ie: `PersonPresenter`



CO-PRESENTED BY  & O'REILLY

**Presenters can work for
specific UI components**

ie: [ProjectWidgetPresenter](#)



CO-PRESENTED BY  & O'REILLY

Example

Move Data From View Into Presenter



CO-PRESENTED BY  & O'REILLY

Example

Move View Logic From View Into Presenter



CO-PRESENTED BY  & O'REILLY

Example

Move Model Logic Into Model



CO-PRESENTED BY  & O'REILLY

Example

Extract Operations Into Service



CO-PRESENTED BY  & O'REILLY

What is a Service?

An operation that stands alone with no encapsulated state.



CO-PRESENTED BY  & O'REILLY

```
class ProjectService  
  
  def create_project  
    generate new project logo automatically  
    create the project  
    send out email notifications  
  
    ...  
  end  
  
end
```



CO-PRESENTED BY  & O'REILLY

```
class AccountService  
  
  def transfer_funds(source, target, amount)  
    transaction do  
      source.withdraw(amount)  
      target.deposit(amount)  
    end  
  end  
  
end
```



CO-PRESENTED BY  & O'REILLY

Example

Extract Operations Into Service



CO-PRESENTED BY  & O'REILLY

Example

Introduce Result Object



CO-PRESENTED BY  & O'REILLY

Example

Organize Large Model w/Mixins



CO-PRESENTED BY  & O'REILLY

Example

Extract RJS From Inline Update



CO-PRESENTED BY  & O'REILLY

Example

Extract RJS Into Renderer



CO-PRESENTED BY  & O'REILLY

Example

Replace Instance Variable w/Local Variable In Partial



CO-PRESENTED BY  & O'REILLY

Example

Extract Class from Long Method



CO-PRESENTED BY  & O'REILLY

More Rails Refactorings

- Extract Complex Creation Into Factory
- Move Before Filter and Friends Into App. Service
- Move View Data From Controller Into Presenter
- Move View Logic From Controller Into Presenter
- Move Extrinsic Callback Into Observer
- Move View Logic From Model Into Presenter



CO-PRESENTED BY  & O'REILLY

Questions?



CO-PRESENTED BY  & O'REILLY

References / Resources

Books

- **Refactoring: Improving The Design Of Existing Code**, *Martin Fowler*
- **The Pragmatic Programmer**, *Dave Thomas and Andy Hunt*
- **Domain Driven Design:
Tackling The Heart Of Complexity In Software**, *Eric Evans*

Online Resources

- <http://www.refactoring.com>



CO-PRESENTED BY



& O'REILLY

References / Resources

Example Application based on Strac

- **Example Agile Project Management**, *Zach Dennis and Drew Colthorp*
<http://github.com/zdennis/strac/tree/refactoring>

Strac

- **Open Source Agile Project Management**, *Mark VanHolstyn and Zach Dennis*
<http://github.com/mvanholstyn/strac/tree/master>



CO-PRESENTED BY  & O'REILLY

Thanks for attending!