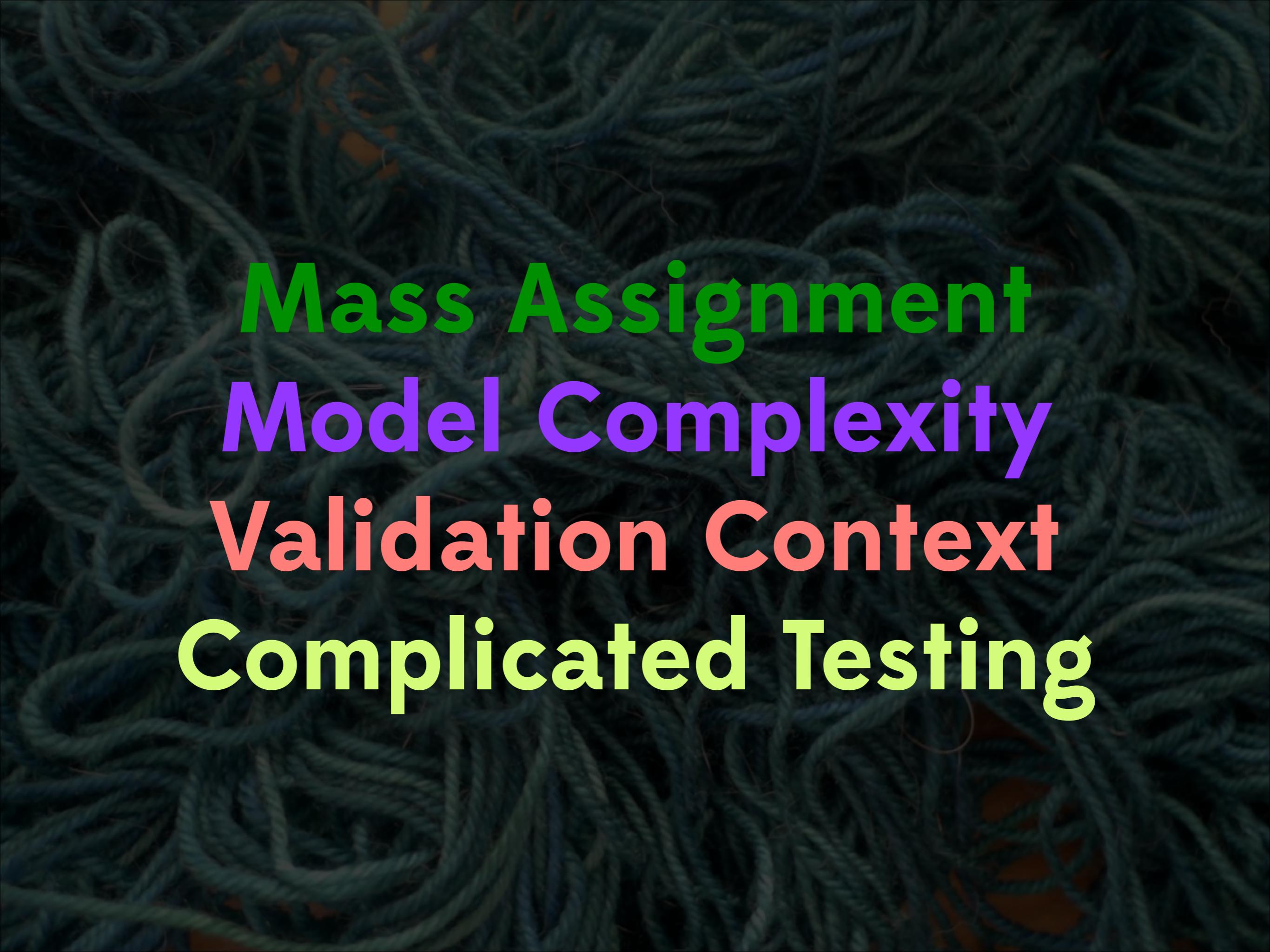


Form Objects & Metrics

Russ Smith - Las Vegas Ruby Group



**Why are form objects
a good idea?**



Mass Assignment
Model Complexity
Validation Context
Complicated Testing

[PERKS](#)[HOURS](#)[PRICING](#)[DIRECTIONS](#)[EVENTS](#)

Get Your Free Day Pass

Ready to come check out the space?

We think you should be able to try it out free of charge. Just fill out the form on your left and we'll send you a free pass via email.

[Email My Free Pass](#)

The Tools



Datamappify
Form Objects
Wicked
MixPanel

```
class DayPass < ActiveRecord::Base  
end
```



No Validations
No Business Logic



ActiveRecord is just a
database wrapper.

A large white truck is shown from a low angle, dumping its load of trash into a massive pile. The truck's side has some red and blue markings. A large flock of birds is flying in the background against a clear sky.

It's not a landfill.

```
class CreateDayPassForm
  include Datamappify::Entity

    attribute :first_name, String
    attribute :last_name, String
    attribute :email, String

    validates :first_name, presence: true
    validates :last_name, presence: true
    validates :email, presence: true
    validate :email_must_be_unique

    def save
      return false unless valid?
      user = User.new
      user.first_name = first_name
      user.last_name = last_name
      user.email = email
      user.save
    end

    private

    def email_must_be_unique
      unless User.where(email: email).count == 0
        errors.add(:email, 'is already taken.')
      end
    end
  end
end
```



So what about the controller?

```
class DayPassesController < ApplicationController
  def new
    @form = CreateDayPassForm.new
  end

  def create
    @form = CreateDayPassForm.new(
      params[:create_day_pass_form] )

    if @form.save
      redirect_to(root_path)
    else
      render(:new)
    end
  end
end
```

```
= simple_form_for(@form,
                  url: day_pass_path,
                  method: :put) do |f|
  = f.input(:first_name, autofocus: true)
  = f.input(:last_name)
  = f.input(:email)
  = f.submit('Email My Free Pass')
```

A close-up photograph of a man's face. He has dark hair and is smiling broadly, showing his teeth. His eyes are slightly squinted. He is wearing a dark, vertically striped shirt. In his right hand, he holds a white smartphone, which is partially visible at the bottom of the frame. The background is a solid, bright blue.

Simple enough so far?

A photograph of a dense forest scene. The foreground is covered in dark, textured ground cover. Numerous tall, thin trees with dark trunks and sparse, light-colored foliage stand in the background, creating a sense of depth.

Multiple Steps

```
class DayPassesController < ApplicationController
  include Wicked::Wizard

  steps :contact_information, :preferences

  def show
    case step
    when :contact_information
      @form = CreateDayPassForm.new
    when :preferences
      @form = PreferencesForm.new
    end

    render_wizard
  end

  def update
    success = case step
    when :contact_information
      @form =
CreateDayPassForm.new(params[:create_day_pass_form])
      @form.save
    when :preferences
      @form = PreferencesForm.new(params[:preferences_form])
      @form.save
    end
    render_wizard(success)
  end

  def finish_wizard_path
    thank_you_path
  end

  # Override Wicked's render_wizard
  def render_wizard(result = nil, options = {})
    if result
      @skip_to ||= @next_step
    else
      @skip_to = nil
    end

    if @skip_to
      redirect_to(wizard_path(@skip_to), options)
    else
      render_step(wizard_value(step), options)
    end
  end
end
```

```
class DayPassesController < ApplicationController
  include Wicked::Wizard

  steps :contact_information, :preferences

  def show
    case step
    when :contact_information
      @form = CreateDayPassForm.new
    when :preferences
      @form = PreferencesForm.new
      @form.customer = current_customer
    end

    render_wizard
  end
end
```

```
class DayPassesController < ApplicationController
  def update
    success = case step
    when :contact_information
      @form =
CreateDayPassForm.new(params[:create_day_pass_form])
      @form.save
    when :preferences
      @form =
PreferencesForm.new(params[:preferences_form])
      @form.customer = current_customer
      @form.save
    end
    render_wizard(success)
  end

  def finish_wizard_path
    thank_you_path
  end
end
```

```
class DayPassesController < ApplicationController
# override wicked's render_wizard
def render_wizard(result = nil, options = {})
  if result
    @skip_to ||= @next_step
  else
    @skip_to = nil
  end

  if @skip_to
    redirect_to(wizard_path(@skip_to), options)
  else
    render_step(wizard_value(step), options)
  end
end
end
```

```
class PreferencesForm
  include Datamappify::Entity

  attr_accessor :customer

  attribute :desk_type, String
  attribute :hours, String

  validates :desk_type, presence: true
  validates :hours, presence: true

  def save
    return false unless valid?

    @customer.desk_type = desk_type
    @customer.hours = hours
    @customer.save
  end
end
```

What's it all doing?

mixpanel
...

```
tracker = Mixpanel::Tracker.new(YOUR_TOKEN)
tracker.track('Contact Information Step', {
  '$email' => current_customer.email
})
```

```
class DayPassesController < ApplicationController
  def update
    success = case step
    when :contact_information
      @form =
CreateDayPassForm.new(params[:create_day_pass_form])
      @form.save
      @tracker.track('Contact Information Step', {
        '$email' => current_customer.email
      })
    when :preferences
      @form = PreferencesForm.new(params[:preferences_form])
      @form.customer = current_customer
      @form.save
      @tracker.track('Preferences Step', {
        '$email' => current_customer.email,
        'Desk Type' => current_customer.desk_type,
        'Hours' => current_customer.hours
      })
    end
  end
end
```

Oct 19, 2013 - Nov 17, 2013 ▾

12.5% Completion Rate



Overview

Subscr...d Step

Subscr...o Step

Subscr...g Step

Subscr...t Step

$$\mathcal{D} = \frac{1}{c} \frac{1}{t} \frac{dl}{dt} = \frac{1}{c} \frac{1}{P} \frac{dP}{dt}$$

$$D^2 = \frac{1}{P^2} \frac{P_0 - P}{P} \sim \frac{1}{P^2} \quad (1a)$$

Lessons Learned

$$D^2 \sim 10^{-53}$$

$$\epsilon \sim 10^{-26}$$

$$P \sim 10^8 L$$

$$t \sim 10^{10} (10^{11})$$

A close-up photograph of a person's hands resting on a dark, textured surface, likely a keyboard or laptop trackpad. A red cloth or piece of fabric is draped over the hands and the surface. The lighting is dramatic, with strong highlights and shadows.

**It's not just enough to
get an app to “work”.**



Multiple steps gives you
the opportunity to measure
your users actions.



**Let metrics drive your
business decisions.**