An Optimistic Proposal for Making Horrible Code... Bearable.

Joe Mastey
RailsConf 2017
Abandon all Hope ye who Enter Here
• Custom system packages (my-libssl)
• 6000+ line User model
• Four days to run test suite, with 10% flapping tests.
• 1,000,000+ lines of ruby
mo platforms, mo problems
our own victims
but why?
we can’t tell what code is even in use anymore
changing code breaks completely unrelated tests
good luck merging your code style PRs
69151 offenses detected
we can’t trust the test suite
so what do they do instead?
make a tiny change
commit it

(skip the tests)
run like hell
learned helplessness
The Trough of Despair

You Are Here
name the evil
badness is unevenly distributed
wc -l $(find . -name *.rb) | sort
... 220 other files here
339 ./lib/devise/models/confirmable.rb
347 ./test/integration/recoverable_test.rb
350 ./test/models/lockable_test.rb
363 ./test/integration/registerable_test.rb
499 ./lib/devise.rb
513 ./lib/devise/rails/routes.rb
519 ./test/models/confirmable_test.rb
698 ./test/integration/authenticatable_test.rb
16262 total
Badness

Number of Files
bundle exec rake stats
<table>
<thead>
<tr>
<th>Name</th>
<th>Lines</th>
<th>LOC</th>
<th>Classes</th>
<th>Methods</th>
<th>M/C</th>
<th>LOC/M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controllers</td>
<td>3347</td>
<td>2570</td>
<td>71</td>
<td>310</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Helpers</td>
<td>432</td>
<td>347</td>
<td>0</td>
<td>49</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Models</td>
<td>1957</td>
<td>1451</td>
<td>51</td>
<td>200</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Mailers</td>
<td>85</td>
<td>71</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Javascripts</td>
<td>9344</td>
<td>6447</td>
<td>0</td>
<td>724</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Libraries</td>
<td>2196</td>
<td>1532</td>
<td>27</td>
<td>143</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>Controller specs</td>
<td>1675</td>
<td>1237</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>245</td>
</tr>
<tr>
<td>…</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>View specs</td>
<td>283</td>
<td>236</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>24956</td>
<td>18451</td>
<td>153</td>
<td>1449</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

Code LOC: 12418  Test LOC: 6033  Code to Test Ratio: 1:0.5
<table>
<thead>
<tr>
<th>Name</th>
<th>Lines</th>
<th>LOC</th>
<th>Classes</th>
<th>Methods</th>
<th>M/C</th>
<th>LOC/M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controllers</td>
<td>3347</td>
<td>2570</td>
<td>71</td>
<td>310</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Helpers</td>
<td>432</td>
<td>347</td>
<td>0</td>
<td>49</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Models</td>
<td>1957</td>
<td>1451</td>
<td>51</td>
<td>200</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Mailers</td>
<td>85</td>
<td>71</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Javascripts</td>
<td>9344</td>
<td>6447</td>
<td>0</td>
<td>724</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Libraries</td>
<td>2196</td>
<td>1532</td>
<td>27</td>
<td>143</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>Controller specs</td>
<td>1675</td>
<td>1237</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>245</td>
</tr>
<tr>
<td>View specs</td>
<td>283</td>
<td>236</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>24956</td>
<td>18451</td>
<td>153</td>
<td>1449</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

Code LOC: 12418    Test LOC: 6033    Code to Test Ratio: 1:0.5
<table>
<thead>
<tr>
<th>Name</th>
<th>Lines</th>
<th>LOC</th>
<th>Classes</th>
<th>Methods</th>
<th>M/C</th>
<th>LOC/M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controllers</td>
<td>3347</td>
<td>2570</td>
<td>71</td>
<td>310</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Helpers</td>
<td>432</td>
<td>347</td>
<td>0</td>
<td>49</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Models</td>
<td>1957</td>
<td>1451</td>
<td>51</td>
<td>200</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Mailers</td>
<td>85</td>
<td>71</td>
<td>3</td>
<td>9</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Javascripts</td>
<td>9344</td>
<td>6447</td>
<td>0</td>
<td>724</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Libraries</td>
<td>2196</td>
<td>1532</td>
<td>27</td>
<td>143</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>Controller specs</td>
<td>1675</td>
<td>1237</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>245</td>
</tr>
<tr>
<td>View specs</td>
<td>283</td>
<td>236</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>24956</td>
<td>18451</td>
<td>153</td>
<td>1449</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

Code LOC: 12418    Test LOC: 6033    Code to Test Ratio: 1:0.5
or make your own
def methods_per_activerecord_class
    children = ObjectSpace.each_object(Class)
                .select { |klass| klass < ActiveRecord::Base }
    sorted = children.sort_by { |klass| klass.methods(false).count }
    sorted.reverse!
    sorted.map { |klass| "#{klass.methods(false).length} #{klass}" }
end
75 Business,
74 User,
59 Product,
53 Group,
48 JournalEntry,
37 Notification,
30 Order,
30 Subscription,
22 Audited::Audit,
12 CombinedPurchaseRecord,
 8 PgSearch::Document
RSpec Active Record Formatter

Adds a new formatting option to rspec that counts your ActiveRecord queries and object creations.

Why? Because database interaction is really slow, and careless creation of large graphs of objects is a primary cause of insanely slow test suites. This project can help you diagnose where you're doing the most damage so that you can start to fix it.

Usage

```ruby
NotifiesOnDisbursements

#initialize
(03, 16) initializes with data
#
#notify
(03, 16) calls the mailers
(11, 47) summarizes the disbursements as it does
(03, 16) sends an extra email to the owner of
(03, 16) doesn't send if you don't configure t
(03, 18) sends everything
#
#summarize
(03, 16) groups net payout on journal entries
(03, 16) separates groups by ID, despite ident
#disbursement.address
```

You'll also get a summary at the end of your test run:

Finished in 26.03 seconds (files took 1.7 seconds to load)
778 examples, 2 failures, 2 pending, 2482 AR objects, 13789 AR queries
fixing the code
stop the bleeding
I’m... back?
# arrowhead of dooooom
Metrics/BlockNesting:
  Max: 4

# short methods, man...
Metrics/MethodLength:
  Max: 414

Metrics/LineLength:
  Max: 324

# fewer parameters
Metrics/ParameterLists:
  Max: 15
# ratchets_spec.rb

ar_classes = ObjectSpace.each_object(Class)
  .select { |klass| klass < ActiveRecord::Base }

METHOD_THRESHOLD = 505

ar_classes.each do |klass|
  describe klass do
    it "shouldn't have more than #{METHOD_THRESHOLD} methods" do
      expect(klass.methods(false).count).to be < METHOD_THRESHOLD
    end
  end
end
strategic improvements
fix it in pieces
breakage is going to happen
Prioritize

• make your tests better
• eliminate dynamic code
• behead the dragon
• prioritize high churn code
foolproof test fix
delete them
make writing new tests fast
# require 'rails_helper'
require 'spec_helper'
eliminate dynamic code
strategy = user.payroll_strategy.constantize
strategy.new(user, params)
strategy.dispatch
next_state = purchase_lifecycle.next_state
purchase.send(:"mark_as_#{next_state}!"")
define_method
constantize
send
method_missing
eval (?!)

@jmmastey
strategy = case user.payroll_strategy
  when 'biweekly' then BiweeklyPayrollStrategy
  when 'fortnightly' then FortnightlyPayrollStrategy
  when 'monthly' then MonthlyPayrollStrategy
end

strategy.new(user, params)
strategy.dispatch
behead the dragon
class RegistersUserFromFacebook
  attr_reader :token, :user, :errors, :message

  def initialize(token)
    @token = token
    @errors = []
  end

  def call
    # 200 lines of doom here
  end
end
# POST /api/users/create_from_facebook

def create_from_facebook
  service = RegistersUserFromFacebook.new(params[:authentication_token])
  service.call

  @user = service.user

  if service.errors.empty?
    render status: :created, location: logged_in_profile_url
  else
    render json: { message: service.message, errors: service.errors },
               status: :unprocessable_entity
  end
end
wait, wtf
69151 offenses detected
1 file inspected, 36 offenses detected
1. name the evil
2. stop the bleeding
3. strategically improve
4. focus on the process
$1.01^{365} = 34.54$

$0.99^{365} = 0.02$
Other Resources

- Refactoring Ruby (Jay Fields)
- Why Programs Fail: A Guide to Systematic Debugging (Andreas Zeller)
- Service Objects in Ruby
- Gems:
  - rubocop / hound ci
  - rspec-activerecord-formatter
  - bundler-stats

thanks!