# Jenga® Driven Development

# Jenga?

Jogo criado pela Leslie Scott e hoje distribuído pela Hasbro.

# Jenga?

O objetivo é mover peças de camadas intermediárias para o topo sem derrubar a torre.

JENGA IS THE ONLY GAME WE PLAY THAT WORKS AS A PERFECT METAPHOR FOR LIFE.
WHY IS THAT, HEIDEGGER?
IN OTHER GAMES, WE WORK TOWARDS OUTWITTING THE OTHER PLAYERS, ACCUMULATING WEALTH, OR WINNING TEMPORARY PRIZES.
IN JENGA, EVERY MOVE WORKS TOWARDS ONLY A SINGLE DEFINITIVE END: THE DESTRUCTION OF THE GAME. A DESTRUCTION WHICH EXISTS BEYOND AND OUTSIDE OF THE GAME ITSELF, BUT IS NONETHELESS INEVITABLE.

THIS IS WHY JENGA IS UNIQUE AMONG ALL GAMES, IN THAT THE OBJECT OF THE GAME IS TO EXPERIENCE ANXIETY, RATHER THAN PLEASURE. THE ANXIETY OF THE UNCERTAIN, SUDDEN, AND ULTIMATE TERMINATION.
BUT THEN THE QUESTION BECOMES, WHY NOT SIMPLY END THE GAME NOW, AND INTENTIONALLY KNOCK OVER THE TOWER? WHY BOTHER AT ALL?
IF WE CAN'T ANSWER THIS MOST BASIC QUESTION, I DON'T SEE HOW WE CAN DECIDE WHICH BLOCKS TO REMOVE.

# Jenga?

É um jogo divertido. Mas tenso, nervoso, dá ansiedade.

Evitando o

# Jenga® Driven Development

# Olá

João Britto

Plataformatec

britto

noteu

plataformatec

tecnologia e engenharia de software

# Estamos contratando!

http://careers.plataformatec.com.br

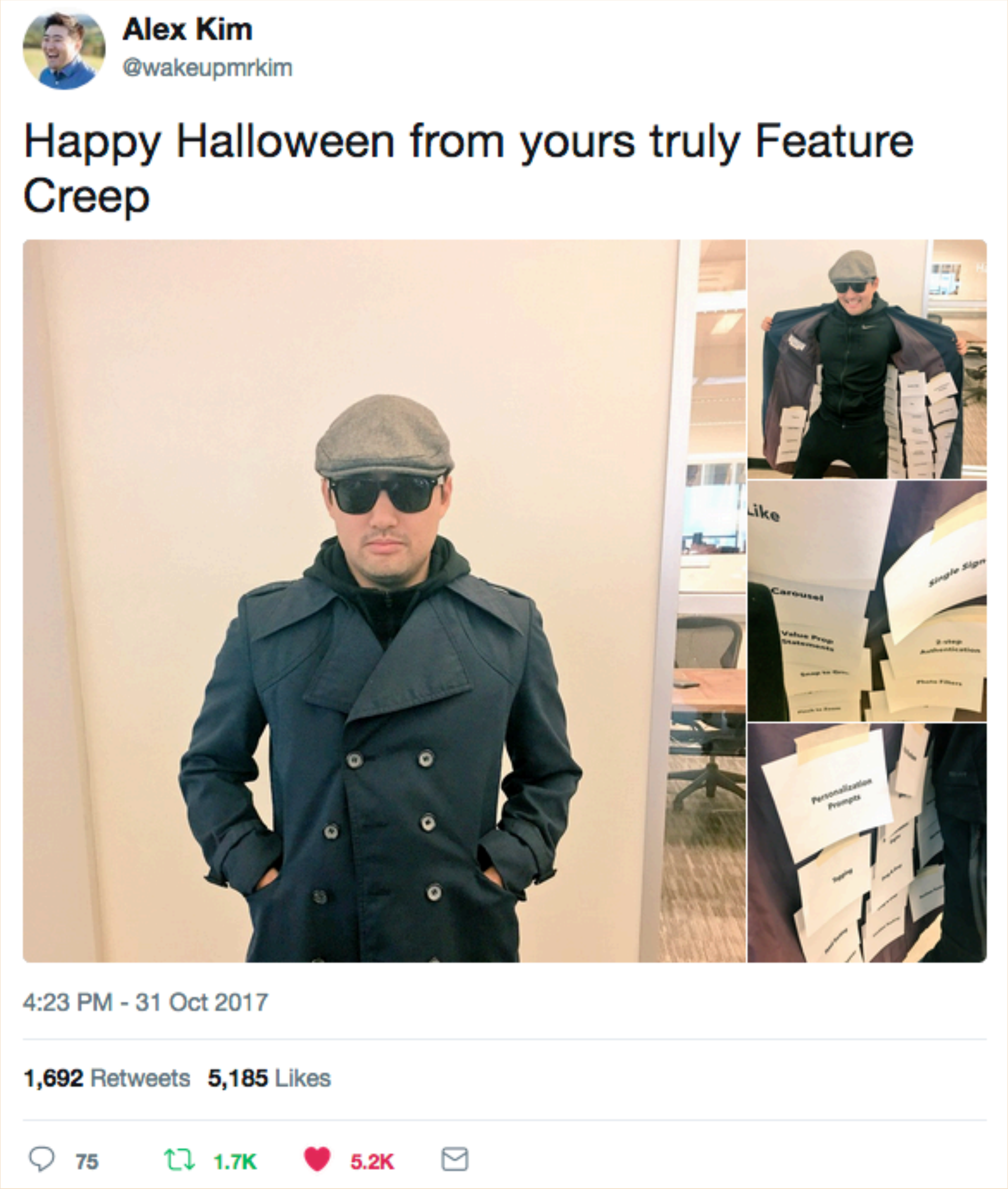
# Codebases

Millions of lines of code

hundred thousand
0
1
2
3
4
5
6
7
8
9
simple iPhone game app
Unix v 1.0
1971
Win32/Simile virus
average iPhone app
Pacemaker
Photoshop v. 1.0
1990
Camino
web browser
Quake 3 engine
3D Video game system
Space Shuttle
a million lines of code
OPERATING SYSTEM
APP
BROWSER
GAME
MACHINE
1
million
1
10
20
30
40
50
60
70
80
90
a million lines of code
18,000 pages of printed text
War And Peace x 14, or
Ulysses x 25, or
The Catcher in The Rye x 63

# Parte 1
# Mo’ Code Mo’ Problems

# Feature Creep



**Alex Kim**
@wakeupmrkim

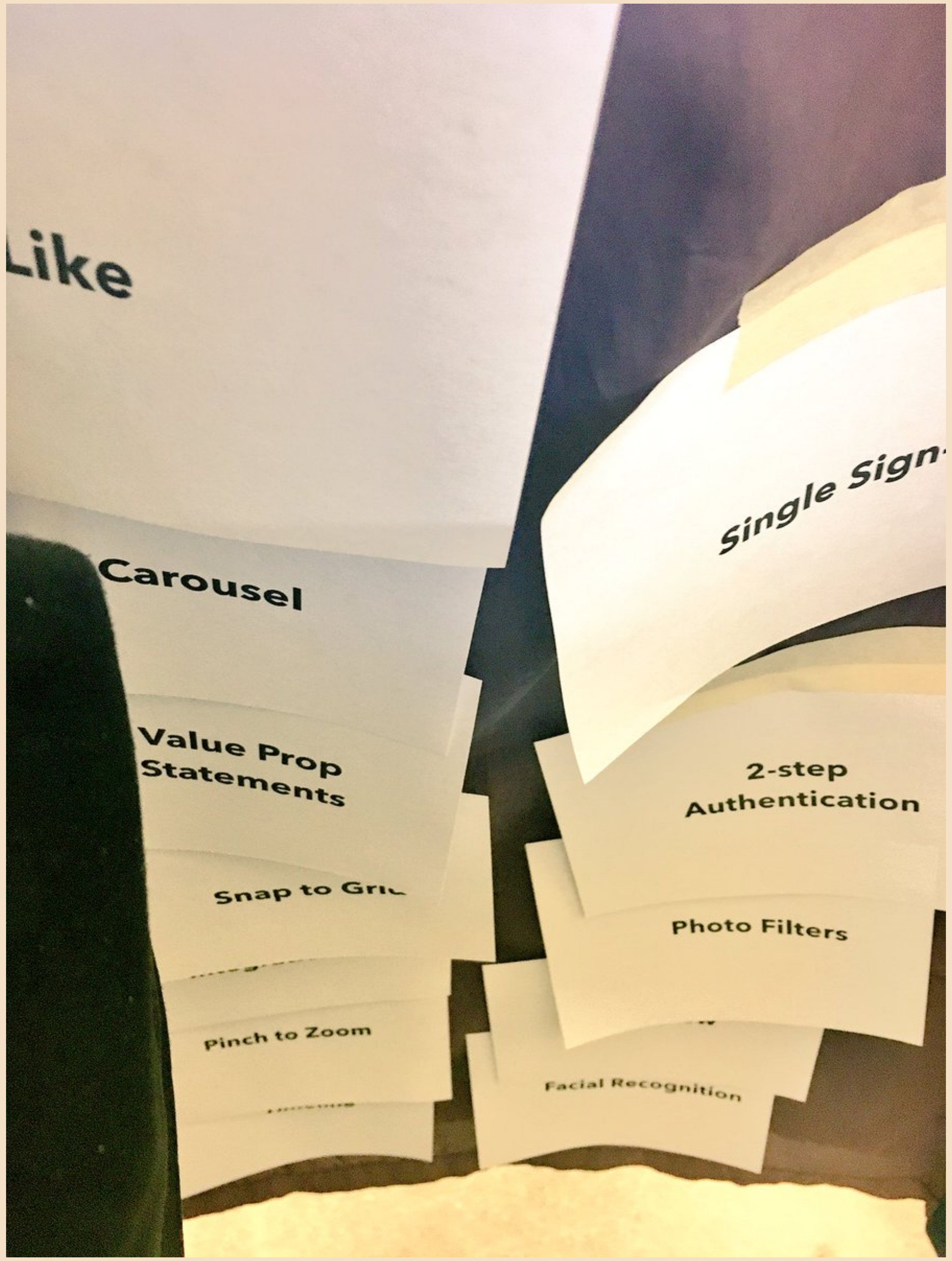
Happy Halloween from yours truly Feature Creep

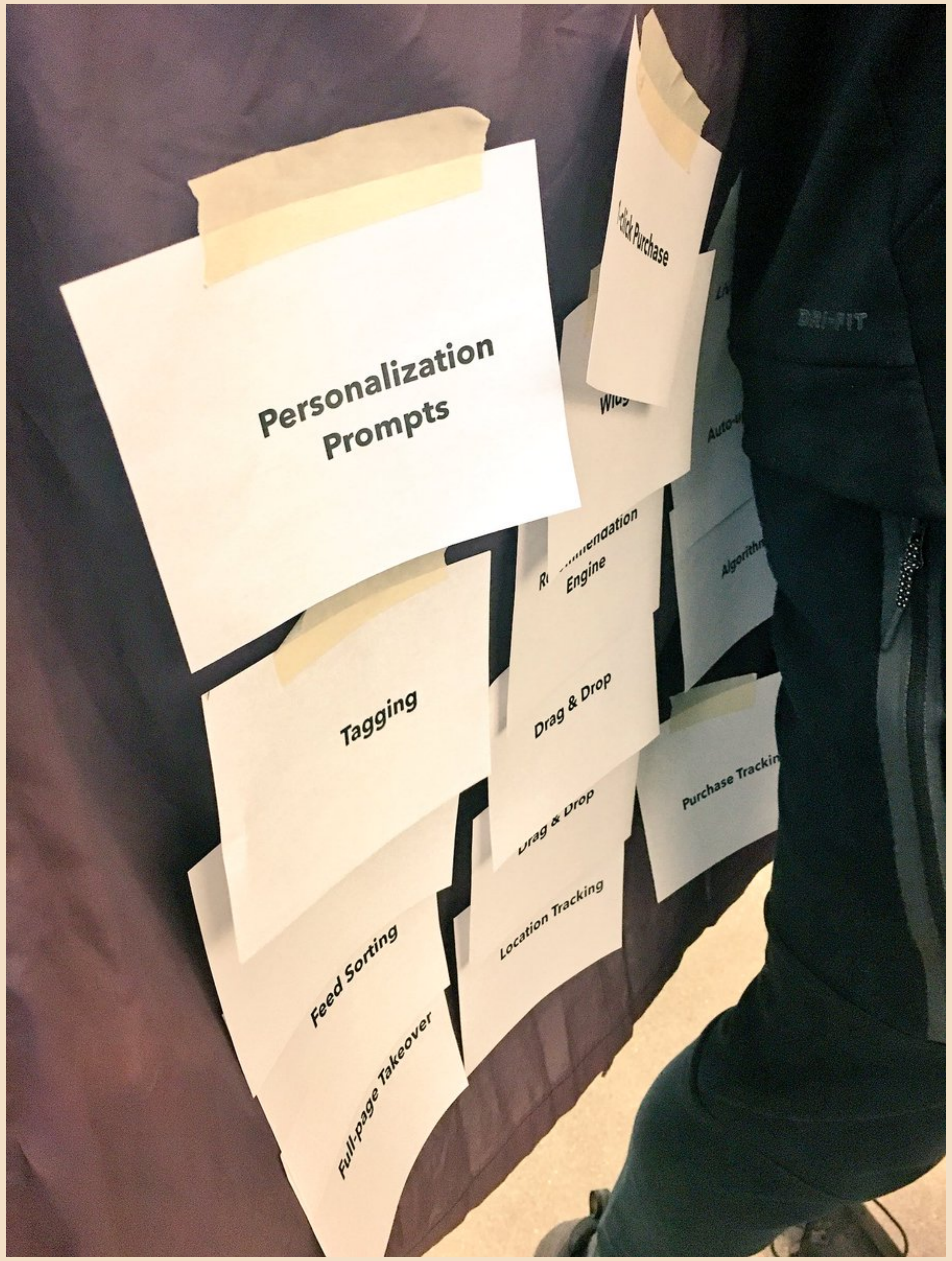
4:23 PM - 31 Oct 2017

**1,692** Retweets **5,185** Likes

75 1.7K 5.2K

# Feature Creep





"A product with lots of features does not make a great product. A great product is one that solves the customer's problem in the simplest way possible. Great products deliver value, not features."

*–Grant Ammons*

**maximizar valor 📈**

**e**

**minimizar custo 📉**

“Whenever you build a new feature, you’re entering into a contract to keep that code up-to-date and compatible with all other features you’ll choose to add in the future.”

*–Sandi MacPherson*

> "Every new line of code you willingly bring into the world is code that has to be debugged, code that has to be read and understood, code that has to be supported. Every time you write new code, you should do so reluctantly, under duress, because you completely exhausted all your other options."

*–Jeff Atwood*

"…if we wish to count lines of code, we should not regard them as 'lines produced' but as 'lines spent'."

*–Edsger W. Dijkstra*

“No code runs faster than no code.
No code has fewer bugs than no code.
No code uses less memory than no code.
No code is easier to understand than no code.”

*–Mike Perham*

No code is easier to delete than no code.

"The easiest code to delete is the code you avoided writing in the first place."

*–Thomas Figg*

# Parte 2
# Código Removível

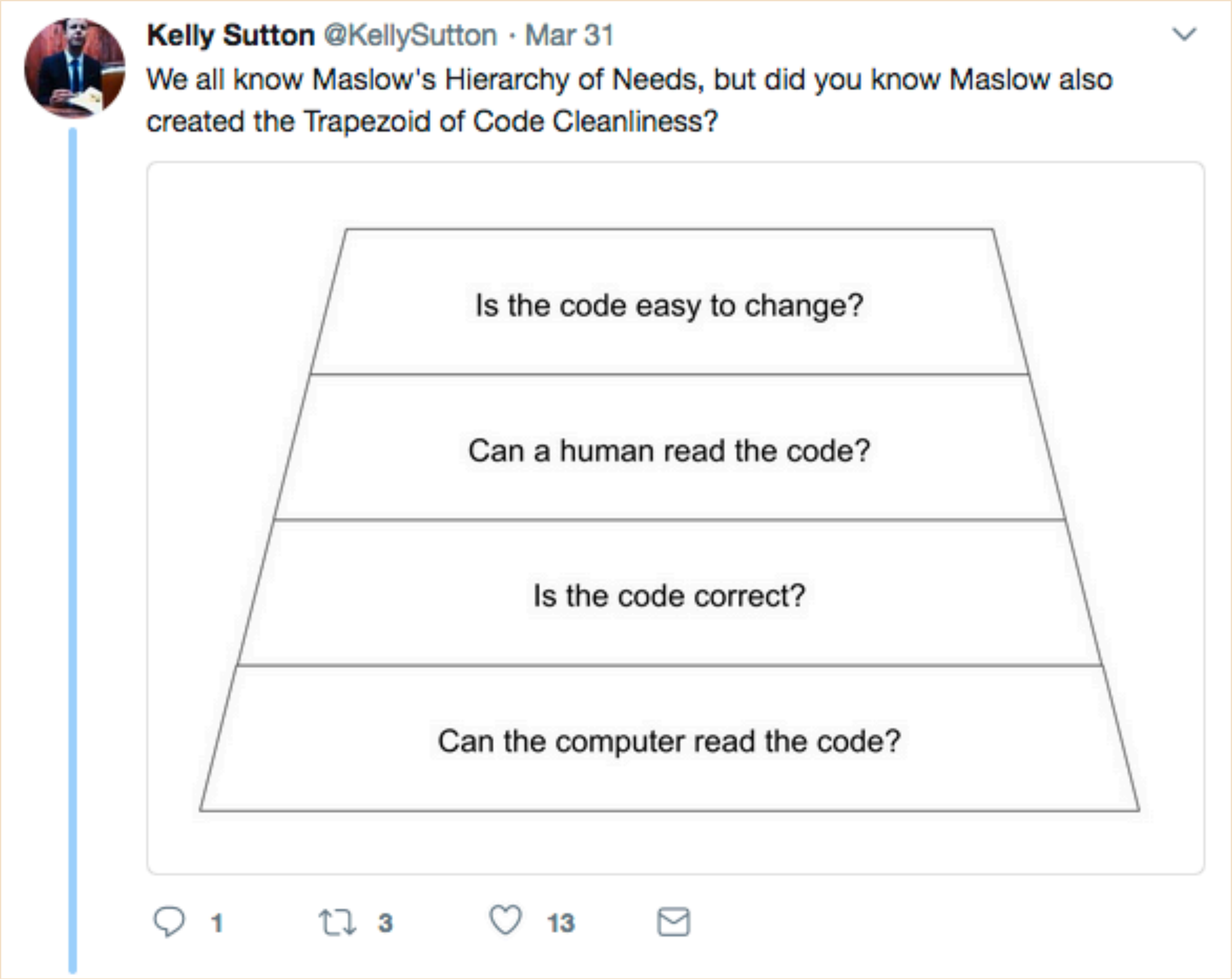
# Coisas mudam

- Funções nascem, crescem e eventualmente morrem.
- Ainda faz sentido no contexto atual do meu produto?
- Tem usuários suficientes para justificar sua existência?
- Representa uma vulnerabilidade na stack? ou seja...
- Está mais atrapalhando do que ajudando?

“The problem with poorly designed *small* applications is that if they are successful they grow up to be poorly designed *big* applications.”

–*Sandi Metz*

# Maslow



**Kelly Sutton** @KellySutton · Mar 31

We all know Maslow's Hierarchy of Needs, but did you know Maslow also created the Trapezoid of Code Cleanliness?

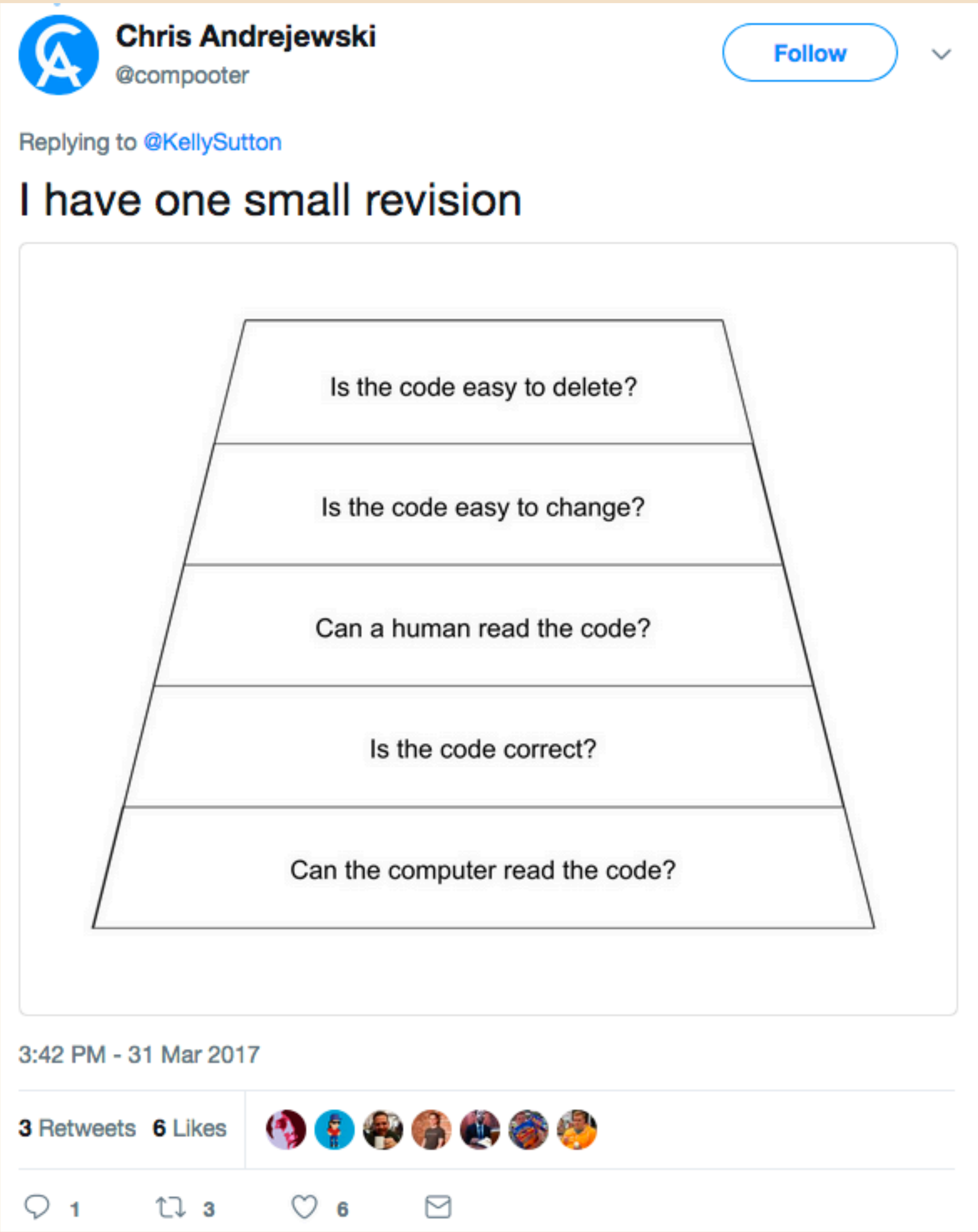
Is the code easy to change?

Can a human read the code?

Is the code correct?

Can the computer read the code?

# Maslow



Chris Andrejewski
@compooter

Follow

Replying to @KellySutton

I have one small revision

Is the code easy to delete?

Is the code easy to change?

Can a human read the code?

Is the code correct?

Can the computer read the code?

3:42 PM - 31 Mar 2017

3 Retweets 6 Likes

1 3 6

# Código pouco “removível”

```ruby
module Animals
  ANIMALS = %i(cat dog fox)

  def self.say(animal)
    if ANIMALS.include?(animal)
      send(animal)
    else
      "What does the #{animal} say?"
    end
  end

  def self.cat; "MEOW!"; end
  def self.dog; "WOOF!"; end
  def self.fox; "WHAT!"; end
end

Animals.say(:dog)
# => "WOOF!"
```

# Código pouco “removível”

```ruby
module Animals
  ANIMALS = %i(cat dog fox)

  def self.say(animal)
    if ANIMALS.include?(animal)
      send(animal)
    else
      "What does the #{animal} say?"
    end
  end

  def self.cat; "MEOW!"; end
  def self.dog; "WOOF!"; end
  def self.fox; "WHAT!"; end
end

Animals.say(:dog)
# => "WOOF!"
```

# Código pouco “removível”

```
module Animals
  ANIMALS = %i(cat dog fox)

  def self.say(animal)
    if ANIMALS.include?(animal)
      send(animal)
    else
      "What does the #{animal} say?"
    end
  end

  def self.cat; "MEOW!"; end

  def self.fox; "WHAT!"; end
end

Animals.say(:dog)
# => NoMethodError: undefined method `dog' for Animals:Module
```

# Código pouco “removível”

```ruby
module Animals
  ANIMALS = %i(cat dog fox)

  def self.say(animal)
    if ANIMALS.include?(animal)
      send(animal)
    else
      "What does the #{animal} say?"
    end
  end

  def self.cat; "MEOW!"; end

  def self.fox; "WHAT!"; end
end

Animals.say(:dog)
# => NoMethodError: undefined method `dog' for Animals:Module
```

# Código pouco “removível”

```
module Animals
  ANIMALS = %i(cat fox)

  def self.say(animal)
    if ANIMALS.include?(animal)
      send(animal)
    else
      "What does the #{animal} say?"
    end
  end

  def self.cat; "MEOW!"; end

  def self.fox; "WHAT!"; end
end

Animals.say(:dog)
# => "What does the dog say?"
```

# Código mais “removível”

```ruby
module Animals
  ANIMALS = {
    cat: "MEOW!",
    dog: "WOOF!",
    fox: "WHAT!"
  }

  def self.say(animal)
    ANIMALS[animal] || "What does the #{animal} say?"
  end
end

Animals.say(:dog)
# => "WOOF!"

Animals.say(:cow)
# => "What does the cow say?"
```

# Código mais “removível”

```ruby
module Animals
  ANIMALS = {
    cat: "MEOW!",
    dog: "WOOF!",
    fox: "WHAT!"
  }

  def self.say(animal)
    ANIMALS[animal] || "What does the #{animal} say?"
  end
end

Animals.say(:dog)
# => "WOOF!"

Animals.say(:cow)
# => "What does the cow say?"
```

# Código mais “removível”

```
module Animals
  ANIMALS = {
    cat: "MEOW!",
    dog: "WOOF!",
    fox: "WHAT!"
  }

  def self.say(animal)
    ANIMALS[animal] || "What does the #{animal} say?"
  end
end

Animals.say(:dog)
# => "WOOF!"

Animals.say(:cow)
# => "What does the cow say?"
```

> “Good code isn’t about getting it right the first time.
> Good code is just legacy code that doesn’t get in the way.”

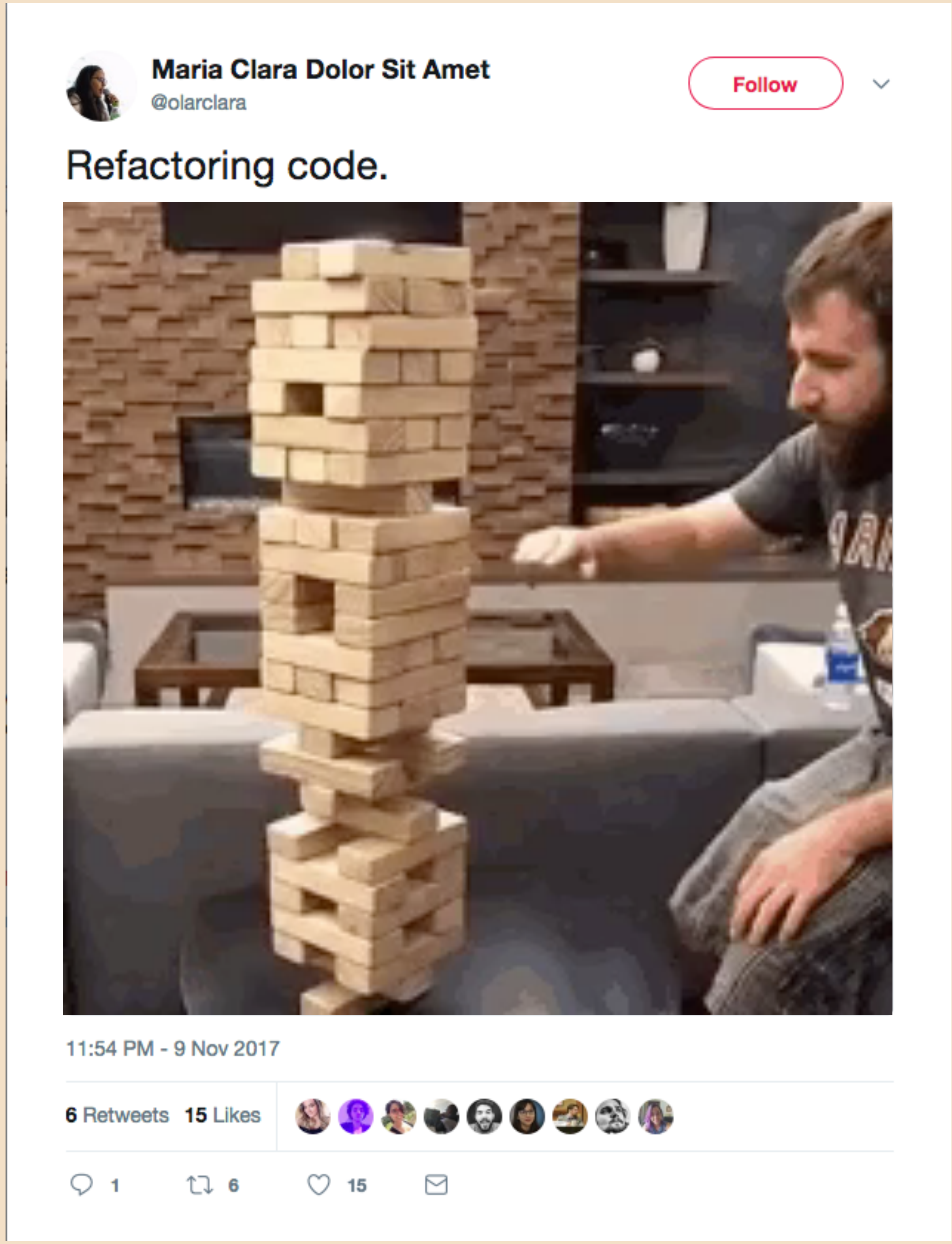
*–Thomas Figg*

# Parte 3
# Removendo Código

# Deveria ser tão simples quanto

1. Apagar o código
2. Apagar os testes
3. Partir pro abraço 😎

# Na prática...



Maria Clara Dolor Sit Amet
@olarclara

Refactoring code.

11:54 PM - 9 Nov 2017

6 Retweets 15 Likes

# Remoções Perigosas

- Inexistente
- Incorreta
- Incompleta

# Remoção Inexistente

# Código intacto

```
def do_stuff(things)
  do_this(things)
  do_that(things)
  do_something_else()
  finish_doing_stuff()
end
```

# Código comentado

```
def do_stuff(things)
  do_this(things)
  # do_that(things)
  # do_something_else()
  finish_doing_stuff()
end
```

HIPPIH
12
1
2
3
4
5
6
7
8
9
10
11
2
3
4
5
6
7
8
9
EXIT

# Código condicionado

```
def do_stuff(things)
  do_this(things)
  if false
    do_that(things)
    do_something_else()
  end
  finish_doing_stuff()
end
```

"…code represents effort expended, and we are very motivated to preserve the value of this effort. And, unfortunately, the sad truth is that the more complicated and incomprehensible the code, i.e. the deeper the investment in creating it, the more we feel pressure to retain it."

*–Sandi Metz*

“Most developers don't like getting rid of stuff. They want to keep chunks of code around in case they need them again. They worked hard to write that chunk of code. They debugged it, it works. They don't want to just throw it away.”

*–Ned Batchelder*

“If you have a chunk of code you don't need any more, there's one big reason to delete it for real rather than leaving it in a disabled state: to reduce noise and uncertainty.”

*–Ned Batchelder*

# Psicologia do controle de versão

git

# Remoções Perigosas

- ~~Inexistente~~
- Incorreta
- Incompleta

# Remoção Incorreta

Jenga

# Remoção Incorreta

```
module Animals
  ANIMALS = %i(cat dog fox)

  def self.say(animal)
    if ANIMALS.include?(animal)
      send(animal)
    else
      "What does the #{animal} say?"
    end
  end

  def self.cat; "MEOW!"; end

  def self.fox; "WHAT!"; end
end

Animals.say(:dog)
# => NoMethodError: undefined method `dog' for Animals:Module
```

“Dependencies are foreign invaders that represent vulnerabilities, and they should be concise, explicit, and isolated.”

*–Sandi Metz*

# Dependências ocultas 👻

# Remoções Perigosas

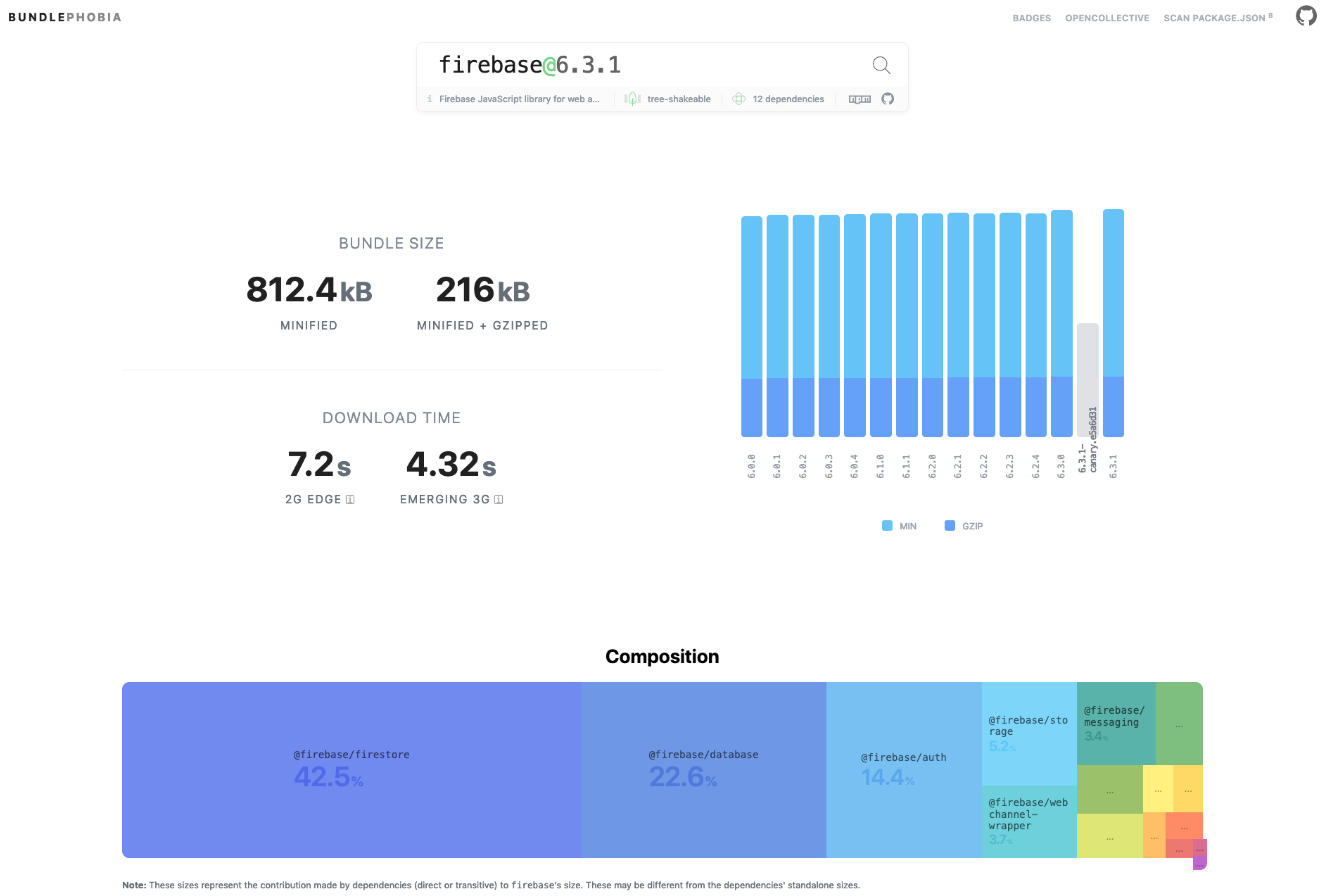
- ~~Inexistente~~
- ~~Incorreta~~
- Incompleta

# Remoção Incompleta

# Sacudir a árvore de dependências

webpack

https://webpack.js.org/guides/tree-shaking/



BUNDLEPHOBIA

BADGES OPENCOLLECTIVE SCAN PACKAGE.JSON B

firebase@6.3.1

Firebase JavaScript library for web a... | tree-shakeable | 12 dependencies

BUNDLE SIZE

812.4 kB MINIFIED

216 kB MINIFIED + GZIPPED

DOWNLOAD TIME

7.2 s 2G EDGE

4.32 s EMERGING 3G

## Composition

**Note:** These sizes represent the contribution made by dependencies (direct or transitive) to `firebase`'s size. These may be different from the dependencies' standalone sizes.

# Purgecss

https://www.purgecss.com/

# Sacolejo Automático

- Reduzir código entregue às máquinas
- Passo de compilação/empacotamento
- Código morto permanece no repositório

# Sacolejo Manual

- Reduzir código entregue às pessoas
- Parte do fluxo de desenvolvimento
- Código removido em definitivo

# Removendo dependências

```
<div class="actions">
  <span class="parting-message">
    <%= goodbye_to(@creature) %>
  </span>

  <%= button_to t('.sacrifice'),
                @creature,
                method: :delete,
                class: 'btn-delete-creature' %>
</div>
```

# Removendo dependências

```
<div class="actions">
  <span class="parting-message">
    <%= goodbye_to(@creature) %>
  </span>

  <%= button_to t('.sacrifice'),
                @creature,
                method: :delete,
                class: 'btn-delete-creature' %>
</div>
```

# Removendo dependências

```
<div class="actions">
  <span class="parting-message">
    <%= goodbye_to(@creature) %>
  </span>

  <%= button_to t('.sacrifice'),
                @creature,
                method: :delete,
                class: 'btn-delete-creature' %>
</div>
```

# Removendo dependências

```
<div class="actions">
  <span class="parting-message">
    <%= goodbye_to(@creature) %>
  </span>

  <%= button_to t('.sacrifice'),
                @creature,
                method: :delete,
                class: 'btn-delete-creature' %>
</div>
```

# Removendo dependências

```
class CreaturesController < ApplicationController
  def destroy
    current_player.creatures.destroy(@creature)

    respond_to do |format|
      format.json { head :no_content }
    end
  end
end
```

# Removendo dependências

```
class CreaturesController < ApplicationController
  def destroy
    current_player.creatures.destroy(@creature)

    respond_to do |format|
      format.json { head :no_content }
    end
  end
end
```

# Removendo dependências

```ruby
class CreaturesController < ApplicationController
  def destroy
    current_player.creatures.destroy(@creature)

    respond_to do |format|
      format.json { head :no_content }
    end
  end
end
```

# Removendo dependências

```
class Player < ApplicationRecord
  has_many :creatures, after_remove: :notify_parents

  def notify_parents(creature)
    NotifyParentsJob.perform_later(creature)
  end
end
```

# Removendo dependências

```
class Player < ApplicationRecord
  has_many :creatures, after_remove: :notify_parents

  def notify_parents(creature)
    NotifyParentsJob.perform_later(creature)
  end
end
```

# Removendo dependências

```ruby
class NotifyParentsJob < ApplicationJob
  queue_as :default

  def perform(creature)
    # Async stuff
  end
end
```

# Removendo dependências

```ruby
class NotifyParentsJob < ApplicationJob
  queue_as :default

  def perform(creature)
    # Async stuff
  end
end
```

# Removendo dependências

```ruby
class Player < ApplicationRecord
  has_many :creatures, after_remove: :notify_parents

  def notify_parents(creature)
    NotifyParentsJob.perform_later(creature)
  end
end
```

# Removendo dependências

```ruby
class Player < ApplicationRecord
  has_many :creatures
end
```

# Removendo dependências

```
class CreaturesController < ApplicationController
  def destroy
    current_player.creatures.destroy(@creature)

    respond_to do |format|
      format.json { head :no_content }
    end
  end
end
```

# Removendo dependências

```
class CreaturesController < ApplicationController
  def destroy
    respond_to do |format|
      format.json { head :no_content }
    end
  end
end
```

# Removendo dependências

```
class CreaturesController < ApplicationController
  def destroy
    respond_to do |format|
      format.json { head :no_content }
    end
  end
end
```

# Removendo dependências

```ruby
class CreaturesController < ApplicationController
end
```

# Removendo dependências

```
<div class="actions">
  <span class="parting-message">
    <%= goodbye_to(@creature) %>
  </span>

  <%= button_to t('.sacrifice'),
                @creature,
                method: :delete,
                class: 'btn-delete-creature' %>
</div>
```

# Removendo dependências

```
<div class="actions">
  <span class="parting-message">
    <%= goodbye_to(@creature) %>
  </span>

  <%= button_to t('.sacrifice'),
                @creature,
                method: :delete,
                class: 'btn-delete-creature' %>
</div>
```

# Removendo dependências

```yaml
en:
  creatures:
    creature:
      disable: "Enter sleep mode"
      enable: "Activate!"
      sacrifice: "Sacrifice creature </3"
```

# Removendo dependências

```
en:
  creatures:
    creature:
      disable: "Enter sleep mode"
      enable: "Activate!"
      sacrifice: "Sacrifice creature </3"
```

# Removendo dependências

```yaml
en:
  creatures:
    creature:
      disable: "Enter sleep mode"
      enable: "Activate!"
```

# Removendo dependências

```
<div class="actions">
  <span class="parting-message">
    <%= goodbye_to(@creature) %>
  </span>

  <%= button_to t('.sacrifice'),
                @creature,
                method: :delete,
                class: 'btn-delete-creature' %>
</div>
```

# Removendo dependências

```
<div class="actions">
  <span class="parting-message">
    <%= goodbye_to(@creature) %>
  </span>

  <%= button_to t('.sacrifice'),
                @creature,
                method: :delete,
                class: 'btn-delete-creature' %>
</div>
```

# Removendo dependências

```
$('.actions').on('click', '.btn-delete-creature', event => {
  this.showCreatureDeletePrompt(event.target);
});
```

# Removendo dependências

```
$('.actions').on('click', '.btn-delete-creature', event => {
  this.showCreatureDeletePrompt(event.target);
});
```

# Removendo dependências

```
$('.actions').on('click', '.btn-delete-creature', event => {
  this.showCreatureDeletePrompt(event.target);
});
```

# Removendo dependências

```
$('.actions').on('click', '.btn-delete-creature', event => {
  this.showCreatureDeletePrompt(event.target);
});
```

# Removendo dependências

```
<div class="actions">
  <span class="parting-message">
    <%= goodbye_to(@creature) %>
  </span>

  <%= button_to t('.sacrifice'),
                @creature,
                method: :delete,
                class: 'btn-delete-creature' %>
</div>
```

# Removendo dependências

```
<div class="actions">
  <span class="parting-message">
    <%= goodbye_to(@creature) %>
  </span>

  <%= button_to t('.sacrifice'),
                @creature,
                method: :delete,
                class: 'btn-delete-creature' %>
</div>
```

# Removendo dependências

```ruby
module CreaturesHelper
  def goodbye_to(creature)
    if creature.enabled?
      I18n.t('creature.goodbye_to.enabled')
    else
      I18n.t('creature.goodbye_to.disabled')
    end
  end
end
```

# Removendo dependências

```ruby
module CreaturesHelper
  def goodbye_to(creature)
    if creature.enabled?
      I18n.t('creature.goodbye_to.enabled')
    else
      I18n.t('creature.goodbye_to.disabled')
    end
  end
end
```

# Removendo dependências

```ruby
module CreaturesHelper
  def goodbye_to(creature)
    if creature.enabled?
      I18n.t('creature.goodbye_to.enabled')
    else
      I18n.t('creature.goodbye_to.disabled')
    end
  end
end
```

# Removendo dependências

```ruby
module CreaturesHelper
  def goodbye_to(creature)
    if creature.enabled?
      I18n.t('creature.goodbye_to.enabled')
    else
      I18n.t('creature.goodbye_to.disabled')
    end
  end
end
```

# Removendo dependências

```
module CreaturesHelper
  def goodbye_to(creature)
    if creature.enabled?
      I18n.t('creature.goodbye_to.enabled')
    else
      I18n.t('creature.goodbye_to.disabled')
    end
  end
end
```

# Removendo dependências

```
<div class="actions">
  <span class="parting-message">
    <%= goodbye_to(@creature) %>
  </span>

  <%= button_to t('.sacrifice'),
                @creature,
                method: :delete,
                class: 'btn-delete-creature' %>
</div>
```

# Removendo dependências

```
<div class="actions">
  <span class="parting-message">
    <%= goodbye_to(@creature) %>
  </span>

  <%= button_to t('.sacrifice'),
                @creature,
                method: :delete,
                class: 'btn-delete-creature' %>
</div>
```

# Removendo dependências

```
.parting-message {
  background-color: #ccc;
}
```

# Removendo dependências

```css
.parting-message {
  background-color: #ccc;
}
```

# Removendo dependências

```
<div class="actions">
  <span class="parting-message">
    <%= goodbye_to(@creature) %>
  </span>

  <%= button_to t('.sacrifice'),
                @creature,
                method: :delete,
                class: 'btn-delete-creature' %>
</div>
```

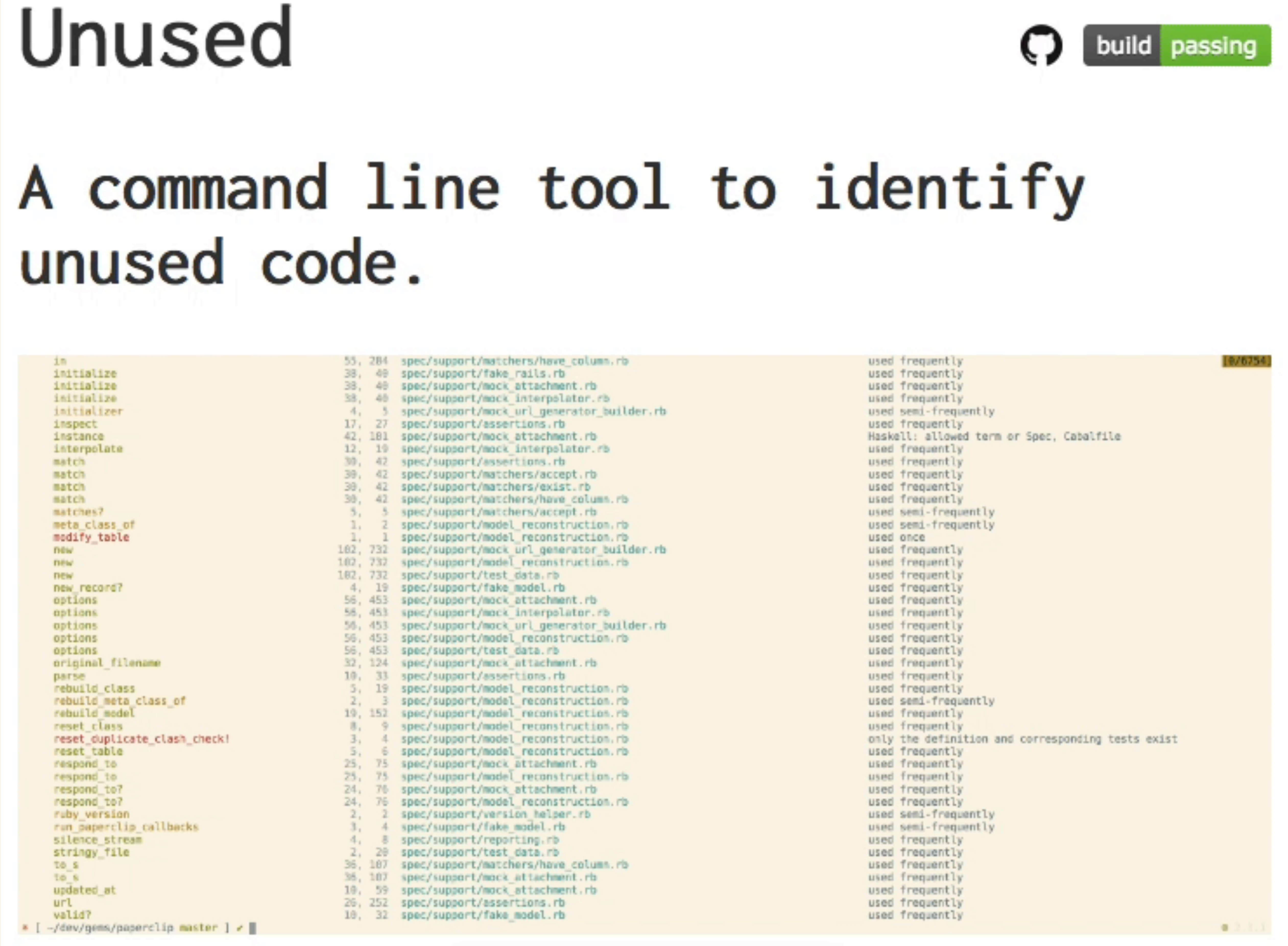
# Removendo dependências

```
<div class="actions">
  <span class="parting-message">
    <%= goodbye_to(@creature) %>
  </span>

  <%= button_to t('.sacrifice'),
                @creature,
                method: :delete,
                class: 'btn-delete-creature' %>
</div>
```

# Feito!

# Ferramentas ajudam



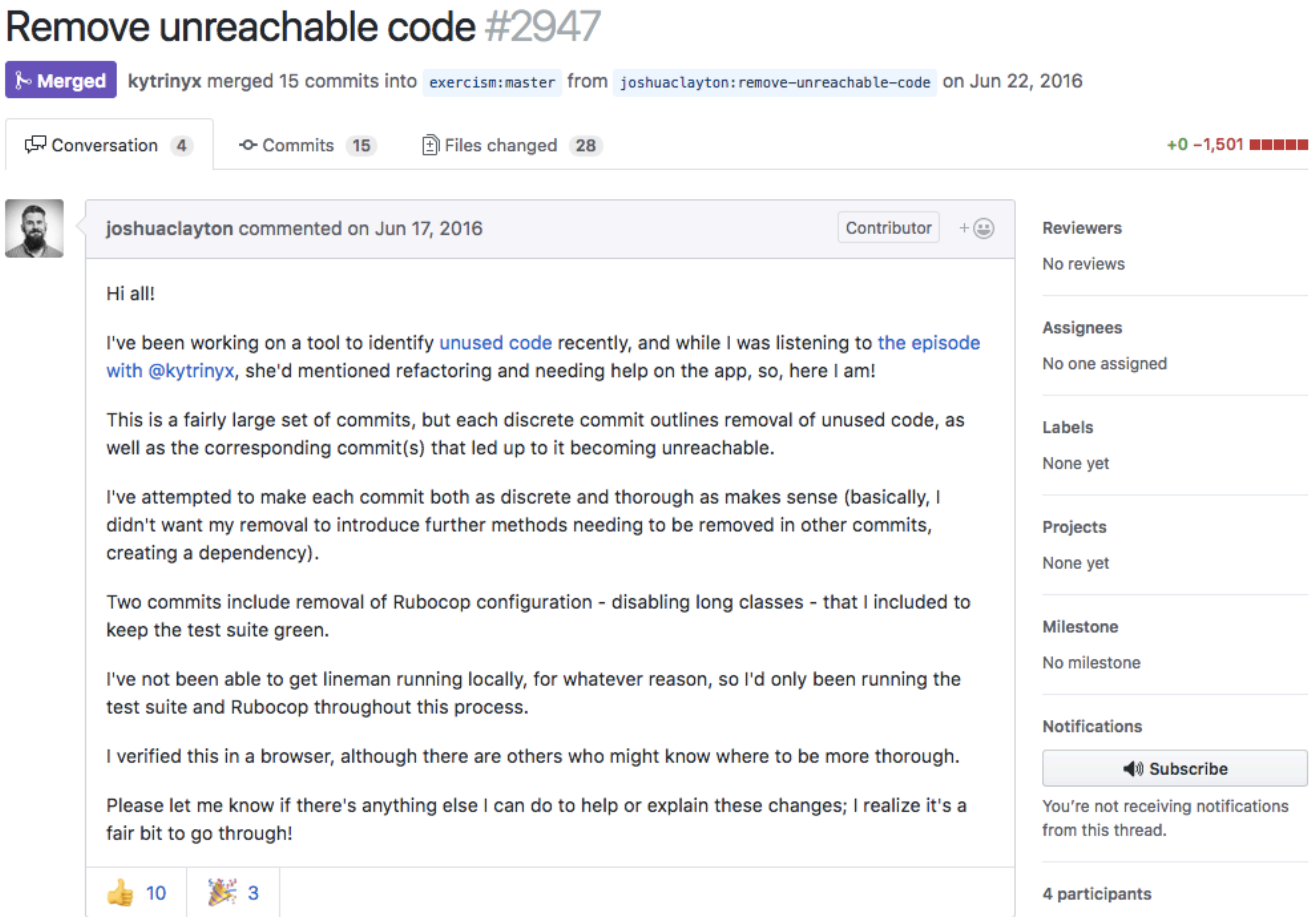
## Unused

A command line tool to identify unused code.

```
    in                                      55, 284  spec/support/matchers/have_column.rb             used frequently                    [0/6754]
    initialize                              38,  49  spec/support/fake_rails.rb                       used frequently
    initialize                              38,  49  spec/support/mock_attachment.rb                  used frequently
    initialize                              38,  49  spec/support/mock_interpolator.rb                used frequently
    initializer                              4,   5  spec/support/mock_url_generator_builder.rb       used semi-frequently
    inspect                                 17,  27  spec/support/assertions.rb                       used frequently
    instance                                42, 181  spec/support/mock_attachment.rb                  Haskell: allowed term or Spec, Cabalfile
    interpolate                             12,  19  spec/support/mock_interpolator.rb                used frequently
    match                                   39,  42  spec/support/assertions.rb                       used frequently
    match                                   39,  42  spec/support/matchers/accept.rb                  used frequently
    match                                   39,  42  spec/support/matchers/exist.rb                   used frequently
    match                                   39,  42  spec/support/matchers/have_column.rb             used frequently
    matches?                                 5,   5  spec/support/matchers/accept.rb                  used semi-frequently
    meta_class_of                            1,   2  spec/support/model_reconstruction.rb             used semi-frequently
    modify_table                             1,   1  spec/support/model_reconstruction.rb             used once
    new                                    102, 732  spec/support/mock_url_generator_builder.rb       used frequently
    new                                    102, 732  spec/support/model_reconstruction.rb             used frequently
    new                                    102, 732  spec/support/test_data.rb                        used frequently
    new_record?                              4,  19  spec/support/fake_model.rb                       used frequently
    options                                 56, 453  spec/support/mock_attachment.rb                  used frequently
    options                                 56, 453  spec/support/mock_interpolator.rb                used frequently
    options                                 56, 453  spec/support/mock_url_generator_builder.rb       used frequently
    options                                 56, 453  spec/support/model_reconstruction.rb             used frequently
    options                                 56, 453  spec/support/test_data.rb                        used frequently
    original_filename                       32, 124  spec/support/mock_attachment.rb                  used frequently
    parse                                   10,  33  spec/support/assertions.rb                       used frequently
    rebuild_class                            5,  19  spec/support/model_reconstruction.rb             used frequently
    rebuild_meta_class_of                    2,   3  spec/support/model_reconstruction.rb             used semi-frequently
    rebuild_model                           19, 152  spec/support/model_reconstruction.rb             used frequently
    reset_class                              8,   9  spec/support/model_reconstruction.rb             used frequently
    reset_duplicate_clash_check!             3,   4  spec/support/model_reconstruction.rb             only the definition and corresponding tests exist
    reset_table                              5,   6  spec/support/model_reconstruction.rb             used frequently
    respond_to                              25,  75  spec/support/mock_attachment.rb                  used frequently
    respond_to                              25,  75  spec/support/model_reconstruction.rb             used frequently
    respond_to?                             24,  76  spec/support/mock_attachment.rb                  used frequently
    respond_to?                             24,  76  spec/support/model_reconstruction.rb             used frequently
    ruby_version                             2,   2  spec/support/version_helper.rb                   used semi-frequently
    run_paperclip_callbacks                  3,   4  spec/support/fake_model.rb                       used semi-frequently
    silence_stream                           4,   8  spec/support/reporting.rb                        used frequently
    stringy_file                             2,  20  spec/support/test_data.rb                        used frequently
    to_s                                    36, 107  spec/support/matchers/have_column.rb             used frequently
    to_s                                    36, 107  spec/support/mock_attachment.rb                  used frequently
    updated_at                              10,  59  spec/support/mock_attachment.rb                  used frequently
    url                                     26, 252  spec/support/assertions.rb                       used frequently
    valid?                                  10,  32  spec/support/fake_model.rb                       used frequently
* [ ~/dev/gems/paperclip master ] ✓
```

https://unused.codes/

# Ferramentas ajudam



Remove unreachable code #2947

Merged kytrinyx merged 15 commits into `exercism:master` from `joshuaclayton:remove-unreachable-code` on Jun 22, 2016

joshuaclayton commented on Jun 17, 2016 Contributor

Hi all!

I've been working on a tool to identify unused code recently, and while I was listening to the episode with @kytrinyx, she'd mentioned refactoring and needing help on the app, so, here I am!

This is a fairly large set of commits, but each discrete commit outlines removal of unused code, as well as the corresponding commit(s) that led up to it becoming unreachable.

I've attempted to make each commit both as discrete and thorough as makes sense (basically, I didn't want my removal to introduce further methods needing to be removed in other commits, creating a dependency).

Two commits include removal of Rubocop configuration - disabling long classes - that I included to keep the test suite green.

I've not been able to get lineman running locally, for whatever reason, so I'd only been running the test suite and Rubocop throughout this process.

I verified this in a browser, although there are others who might know where to be more thorough.

Please let me know if there's anything else I can do to help or explain these changes; I realize it's a fair bit to go through!

Reviewers
No reviews

Assignees
No one assigned

Labels
None yet

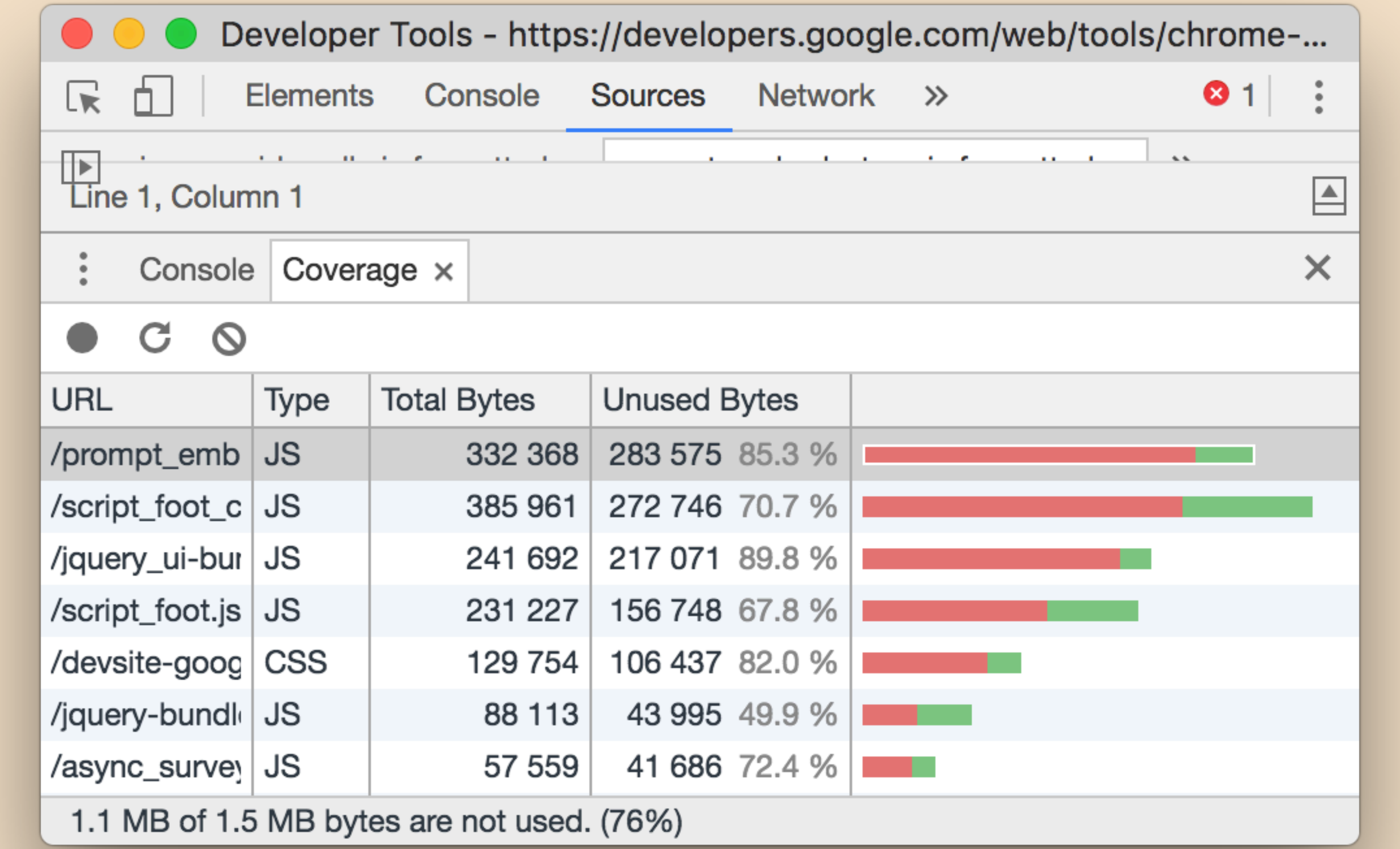
Projects
None yet

Milestone
No milestone

Notifications
Subscribe
You're not receiving notifications from this thread.

4 participants

# Ferramentas ajudam


Developer Tools - https://developers.google.com/web/tools/chrome-...
Elements
Console
Sources
Network
»
1
Line 1, Column 1
Console
Coverage ×
×

| URL | Type | Total Bytes | Unused Bytes |
|---|---|---|---|
| /prompt_emb | JS | 332 368 | 283 575 85.3 % |
| /script_foot_c | JS | 385 961 | 272 746 70.7 % |
| /jquery_ui-bu | JS | 241 692 | 217 071 89.8 % |
| /script_foot.js | JS | 231 227 | 156 748 67.8 % |
| /devsite-goog | CSS | 129 754 | 106 437 82.0 % |
| /jquery-bundl | JS | 88 113 | 43 995 49.9 % |
| /async_survey | JS | 57 559 | 41 686 72.4 % |

1.1 MB of 1.5 MB bytes are not used. (76%)

# Checklist das dependências

- CSS, JavaScript, imagens, assets em geral
- Chaves de cache de longa expiração
- Tarefas assíncronas
- Gemfile
- Arquivos de configuração, I18n
- Documentação, README, etc.
- Receitas de provisionamento
- Banco de dados: migrações, tabelas, colunas, views, triggers, etc.
- Testes: shared examples, helpers, etc.

# CSS, JavaScript, Imagens…

# Chaves de cache

# Tarefas assíncronas

# Bibliotecas externas e Gemfile

# Arquivos de configuração e outros

# Documentação, README

# Receitas de provisionamento

# Banco de dados

- Colunas
- Índices
- Tabelas
- Views
- Triggers
- Migrações

# Testes

- Shared Examples
- Shared Helpers
- Factories
- Fixtures

# Checklist das dependências

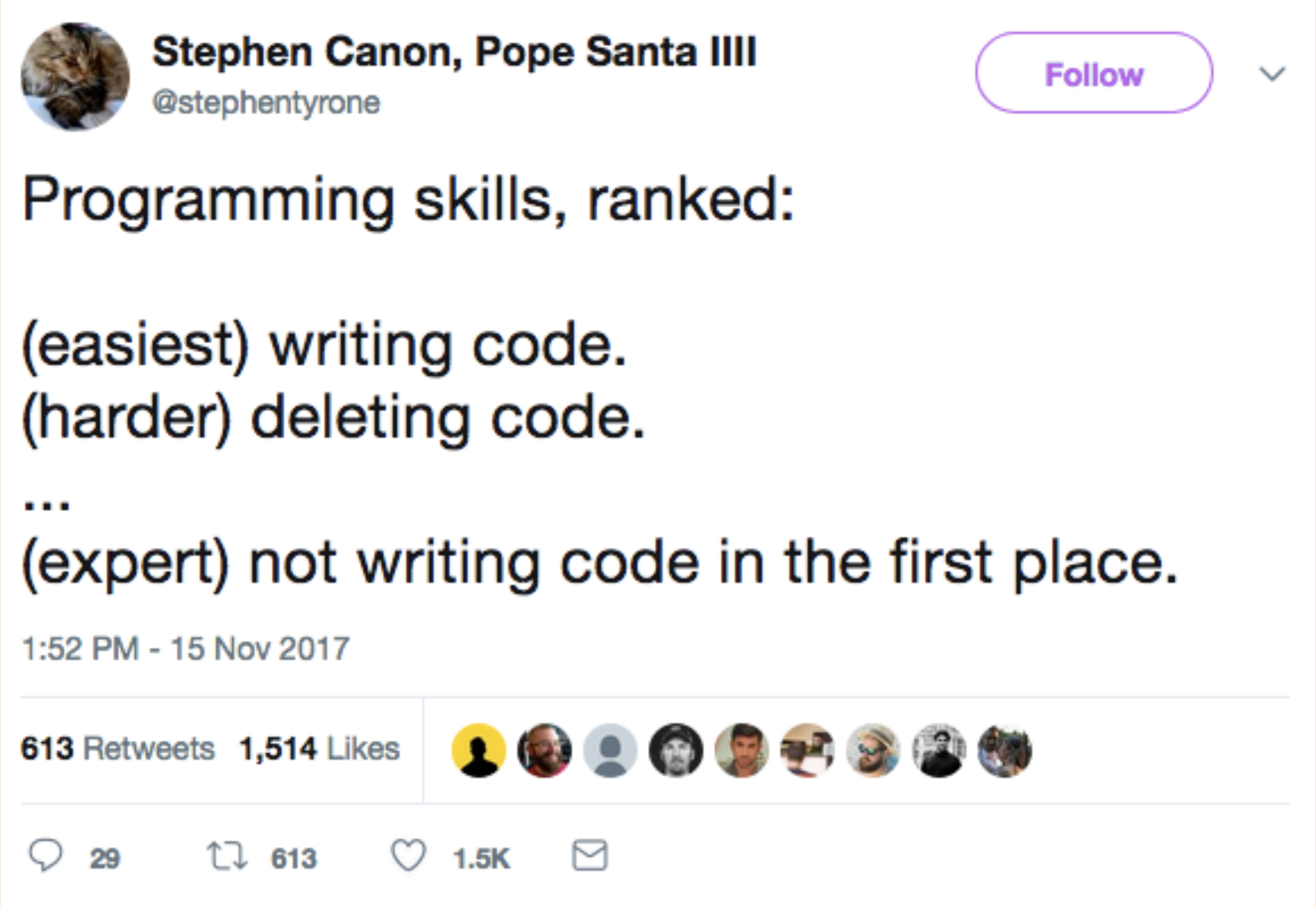
- CSS, JavaScript, Imagens, assets em geral
- Chaves de cache de longa expiração
- Tarefas assíncronas
- Gemfile
- Arquivos de configuração, I18n
- Documentação, README, etc.
- Receitas de provisionamento
- Banco de dados: migrações, tabelas, colunas, views, triggers, etc.
- Testes: shared examples, helpers, etc.

Código tem preço,
menos geralmente é mais

# Fazer mais com menos

- Usando abstrações adequadas e produtivas
- Promovendo reuso através de bibliotecas
- Pensando melhor antes de atacar os problemas
- Funcionalidades equilibradas, sem rebarbas
- Avaliando valor de funcionalidades e produtos

Escreva seu código pensando
no dia de apagá-lo



**Stephen Canon, Pope Santa IIII**
@stephentyrone

Programming skills, ranked:

(easiest) writing code.
(harder) deleting code.
...
(expert) not writing code in the first place.

1:52 PM - 15 Nov 2017

613 Retweets 1,514 Likes

# Remover código não é tão simples quanto parece

“Any code you can delete is a victory, even if you wrote it just yesterday. Do not mourn the loss of time. Celebrate the reduction in long term cost of maintenance.”

*–Eric Normand*

# Obrigado!

britto

noteu

# Fontes e referências

- The Pink Panther in “Pink Outs”
  - https://youtu.be/1q2hu-mDtKs
- Jenga®
  - https://en.wikipedia.org/wiki/Jenga
- Existential Comics - Jenga and the Meaning of Life
  - http://existentialcomics.com/comic/226
- Information is beautiful - Codebases
  - http://www.informationisbeautiful.net/visualizations/million-lines-of-code/
- Feature Creep
  - https://twitter.com/wakeupmrkim/status/925427903352922114
- Grant Ammons - Killing features — just as important as building them
  - https://medium.com/@gammons/killing-features-just-as-important-as-building-them-7f4d64223585

# Fontes e referências

- Sandi MacPherson - Tough tradeoffs: Removing product features is hard, but often necessary
  - https://thenextweb.com/dd/2015/01/18/tough-tradeoffs-removing-product-features-hard-often-necessary/
- Jeff Atwood - The Best Code is No Code At All
  - https://blog.codinghorror.com/the-best-code-is-no-code-at-all/
- Edsger W. Dijkstra - EWD 1036
  - https://www.cs.utexas.edu/~EWD/transcriptions/EWD10xx/EWD1036.html
- Mike Perham - Kill Your Dependencies
  - http://www.mikeperham.com/2016/02/09/kill-your-dependencies/
- Thomas Figg - Write code that is easy to delete, not easy to extend.
  - https://programmingisterrible.com/post/139222674273/write-code-that-is-easy-to-delete-not-easy-to
- Sandi Metz - Practical Object-Oriented Design in Ruby
  - http://www.poodr.com/

# Fontes e referências

- Maslow's Trapezoid of Code Cleaniness
  - https://twitter.com/KellySutton/status/847875703383117824
  - https://twitter.com/compooter/status/847881828224684032
- Refactoring code
  - https://twitter.com/olarclara/status/928802983130927104
- Ned Batchelder - Deleting code
  - https://nedbatchelder.com/text/deleting-code.html
- Tree Shaking
  - http://gph.is/2pPEHXx
- Webpack - Tree Shaking
  - https://webpack.js.org/guides/tree-shaking/
- BundlePhobia
  - https://bundlephobia.com/

# Fontes e referências

- Purgecss
  - https://www.purgecss.com/
- Unused
  - https://unused.codes/
- Google Chrome Dev Tools Code Coverage Tab
  - https://developers.google.com/web/updates/2017/04/devtools-release-notes#coverage
- Programming skills
  - https://twitter.com/stephentyrone/status/930825774818455553
- From 10x programmer to 0.1x programmer: creating more with less
  - https://codewithoutrules.com/2016/08/25/the-01x-programmer/
- Celebrate the reduction in long term cost of maintenance
  - https://twitter.com/ericnormand/status/999303254772105222
- Imagens e vídeos de arquivo pessoal