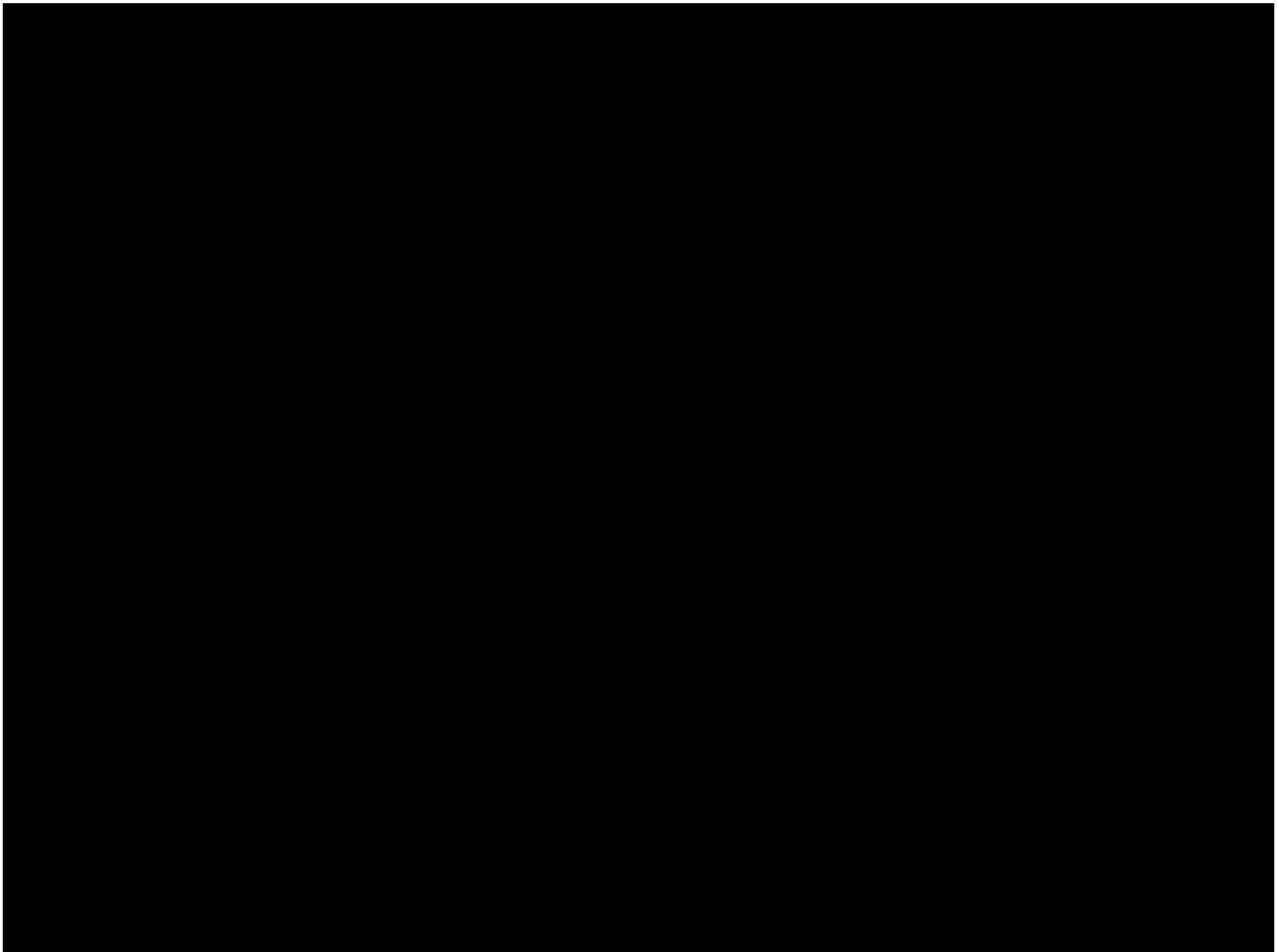
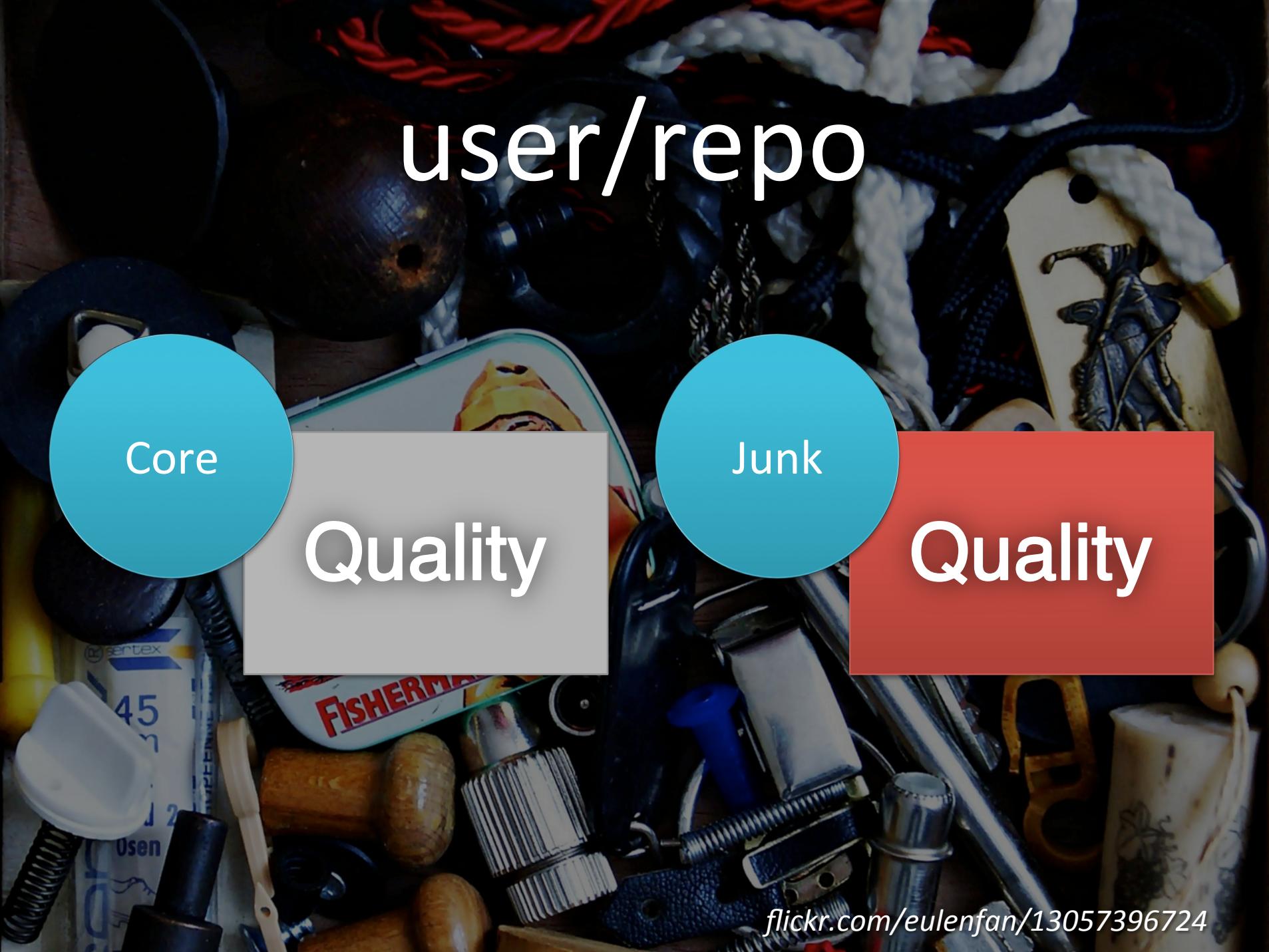


JUNK



flickr.com/calliope/9070783026





user/repo

Core

Junk

Quality

Quality

mitchellh/vagrant

Core

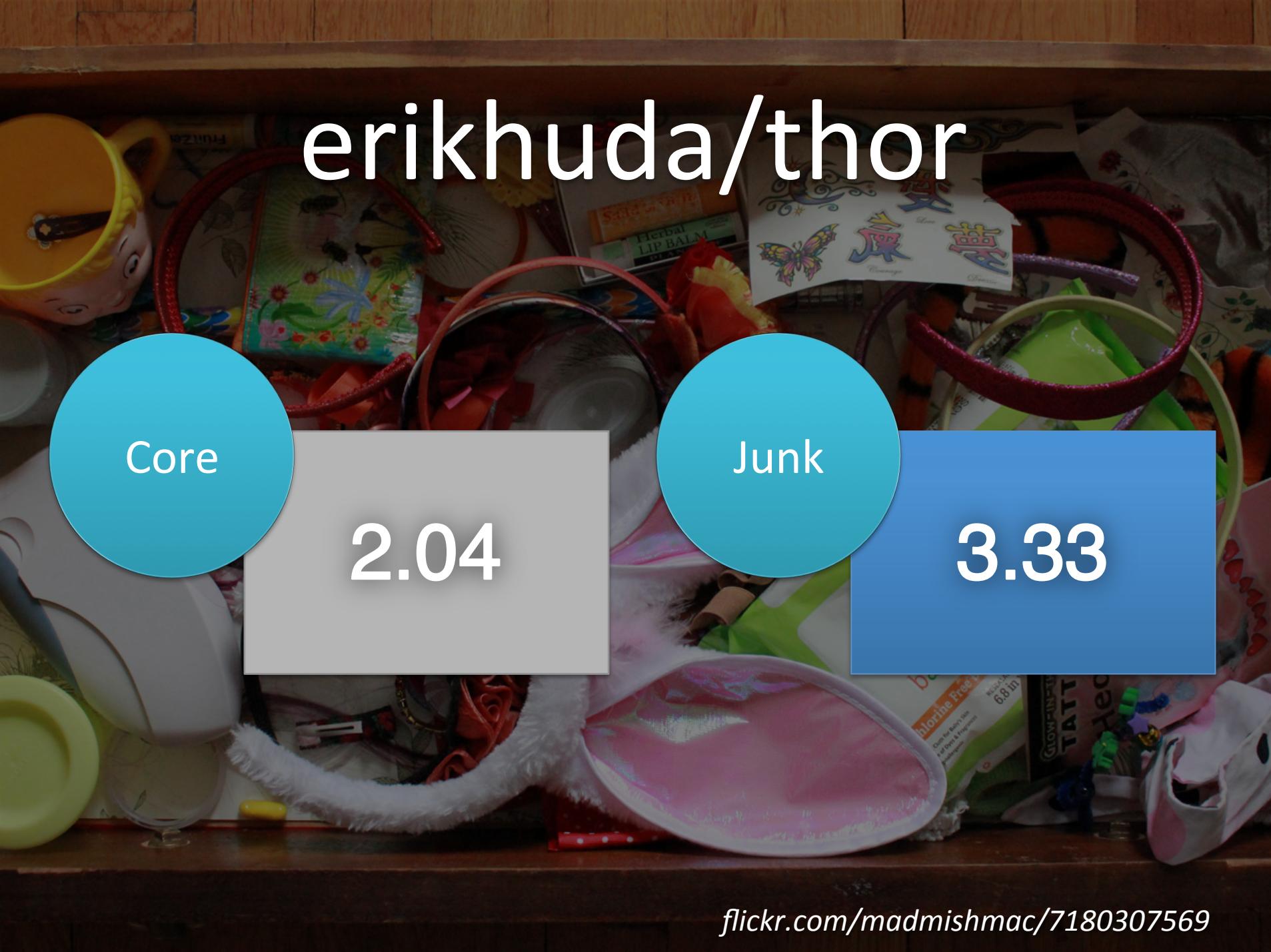
3.62

Junk

3.21

flickr.com/11746801@N04/12761961524

erikhuda/thor



Core

2.04

Junk

3.33

flickr.com/madmishmac/7180307569

sass/sass

Core

2.37

Junk

2.5

[flickr.com/madmishmac/7180306927](https://flickr.com/photos/madmishmac/7180306927)

test-kitchen/test-kitchen

Core

3.44

Junk

3.28

flickr.com/willowherb/2482229479

Average All Repos

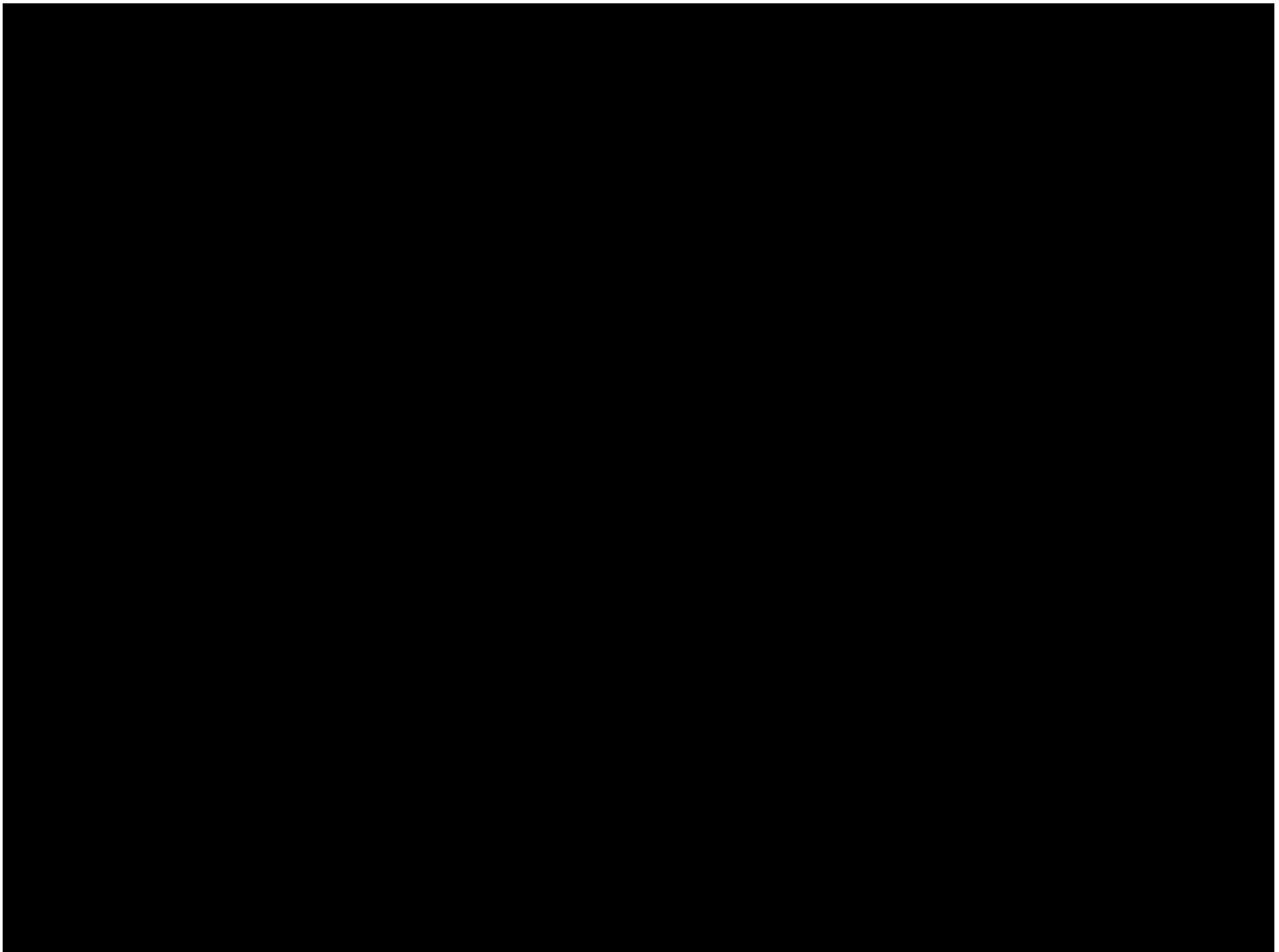
Core

3.11

Junk

3.47

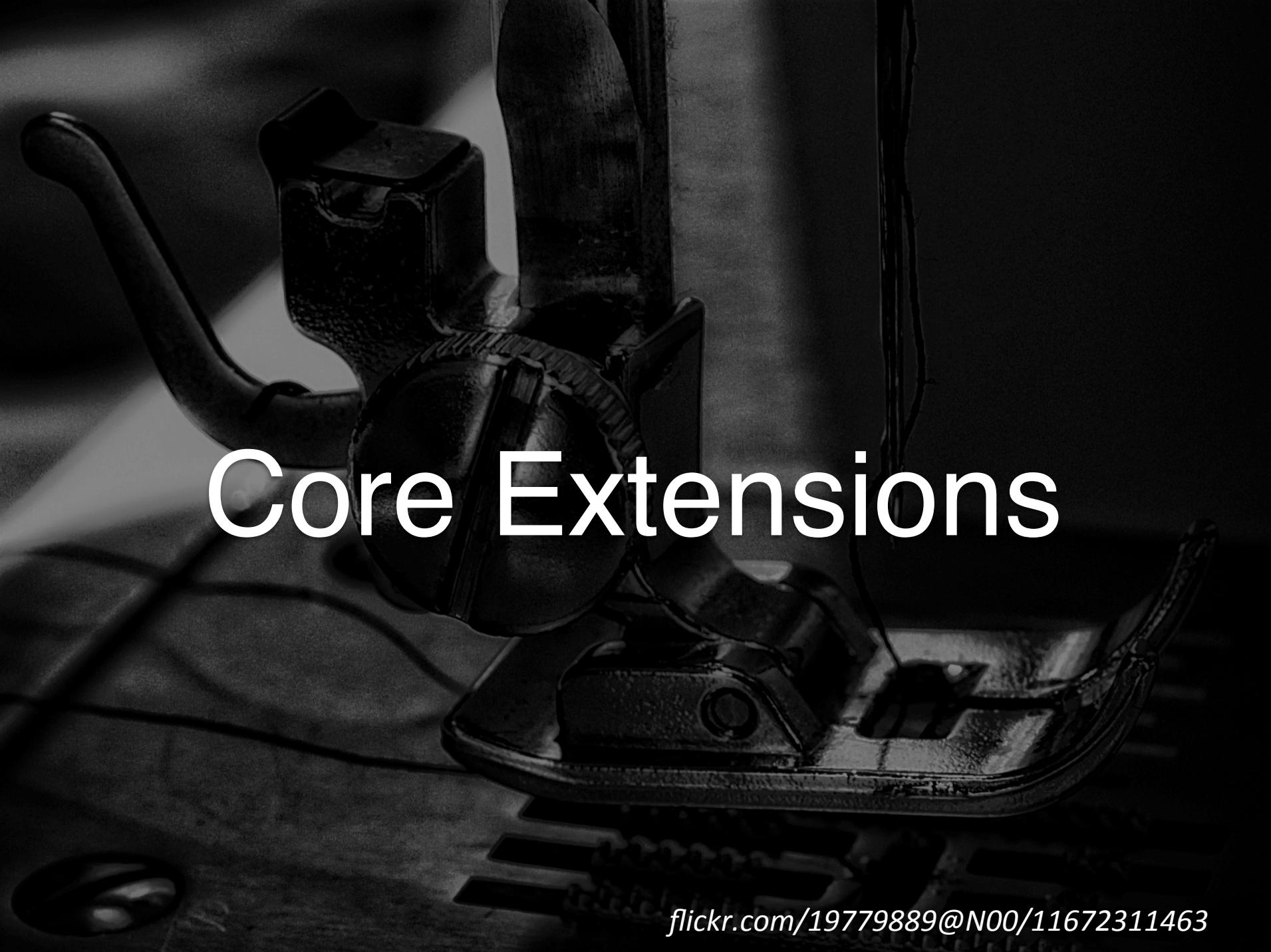
flickr.com/madmishmac/7365534566





Treasure

[flickr.com/19779889@N00/13065138333](https://flickr.com/photos/19779889@N00/13065138333)



Core Extensions

flickr.com/19779889@N00/11672311463



Backwards Compatibility

flickr.com/19779889@N00/12593276135

HashWithIndifferentAccess

flickr.com/19779889@N00/13087691523

Question

mitchellh/vagrant



```
busy.rb
```

```
1 module Vagrant
2   module Util
3     # Utility class which allows blocks of code to be
4     # with a specified interrupt handler. During busy
5     # is often undesirable for SIGINTs to immediately
6     # This class is a helper to cleanly register call
7     # situation.
8     class Busy
9       @@registered = []
10      @@mutex = Mutex.new
11
12      class << self
13        # Mark a given block of code as a "busy" block
14        # register a SIGINT handler for the duration
15        # SIGINT occurs, the `sig_callback` proc will
16        # to the callback to behave properly and exit
```

Line 38, Column 33 Spaces: 2 Ruby

mitchellh/vagrant



```
busy_example.rb
```

```
7 close_connection = lambda do
8   puts "Closing Connection!"
9   @connected = false
10 end
11
12 Vagrant::Util::Busy.busy(close_connection) do
13
14   open_connection!
15   while connected
16     print "> "
17     input = STDIN.gets
18     execute(input)
19   end
20 end
21
```

Line 21, Column 1 Spaces: 2 Ruby

rails/rails



A screenshot of a Mac OS X desktop showing a code editor window. The window title is "watch_stack_example_00.rb — code". The tab bar shows five files: "watch_stack_example_01.rb", "watch_stack_example_00.rb", "watch_stack_example_02.rb", "watch_stack_example_03.rb", and "watch_stack_example_04.rb". The "watch_stack_example_00.rb" tab is selected. The code editor displays the following Ruby code:

```
1 require 'active_support'  
2  
3 ActiveSupport::Dependencies::WatchStack  
4  
5 # Prepare Watcher  
6  
7 # Load the Files  
8  
9 # Mark Constants as Unloadable
```

The status bar at the bottom left says "Line 3, Column 40". The status bar at the bottom right says "Spaces: 2" and "Ruby".

rails/rails



A screenshot of a Mac OS X desktop showing a code editor window. The window title is "watch_stack_example_01.rb — code". The tab bar shows five files: "watch_stack_example_01.rb", "watch_stack_example_00.rb", "watch_stack_example_02.rb", "watch_stack_example_03.rb", and "watch_stack_example_04.rb". The main pane displays Ruby code:

```
1 require 'active_support'  
2  
3 # Prepare Watcher  
4  
5 def watcher  
6   @watcher ||= ActiveSupport::Dependencies::  
7     WatchStack.new  
8 end  
9  
9 def prepare_watcher!  
10  ActiveSupport::Dependencies.clear  
11  watcher.watch_namespaces([ Object ])  
12 end
```

The code editor interface includes standard OS X window controls (red, yellow, green buttons) and status bars at the bottom indicating "Line 5, Column 12", "Spaces: 2", and "Ruby".

rails/rails



A screenshot of a Mac OS X desktop showing a code editor window titled "watch_stack_example_02.rb — code". The window contains Ruby code for managing a file stack. The code includes methods for loading files and paths, utilizing the Dir module and \$LOAD_PATH. The code editor interface shows tabs for other files like "watch_stack_example_01.rb", "watch_stack_example_00.rb", "watch_stack_example_03.rb", and "watch_stack_example_04.rb". The status bar at the bottom indicates "Line 3, Column 18", "Spaces: 2", and "Ruby".

```
13 # Load the files
14
15 def load_files
16   $LOAD_PATH.unshift(Dir.pwd) unless $LOAD_PATH.
17     include?(Dir.pwd)
18   load_path 'lib'
19 end
20
21 def load_path(path,options = {})
22   files = Dir["#{path}/**/*.rb"]
23   files.each {|file| require_or_load file }
24
```

rails/rails



A screenshot of a Mac OS X window titled "watch_stack_example_03.rb — code". The window contains a Ruby script with syntax highlighting. The code is as follows:

```
25 # Mark Constants As Unloadable
26
27 def mark_constants_as_unloadable!
28   watcher.new_constants.each { |constant|
29     unloadable constant
30   }
31 end
```

The line "31" is highlighted with a gray background. The status bar at the bottom shows "Line 31, Column 1" on the left, "Spaces: 2" in the middle, and "Ruby" on the right.

rails/rails

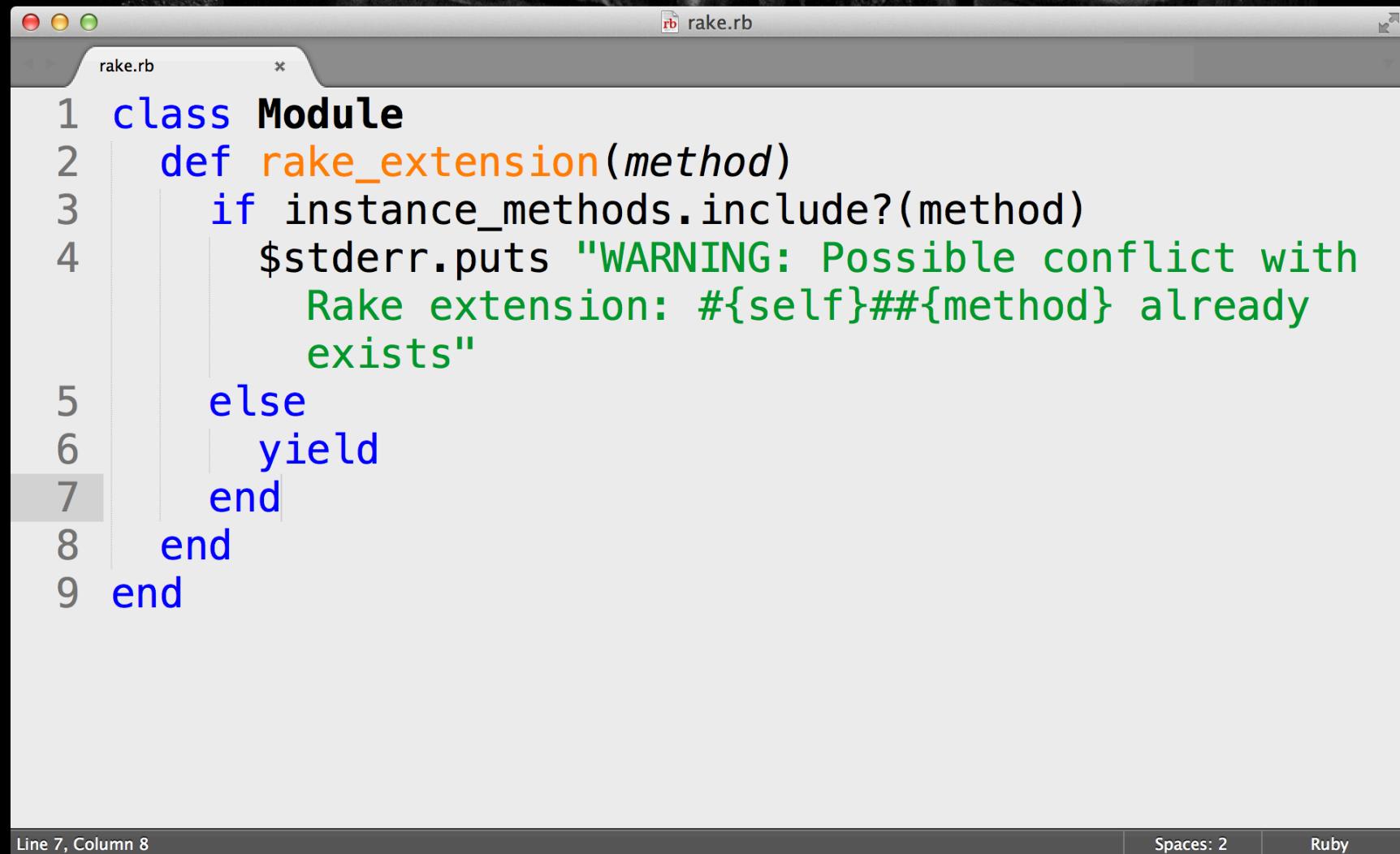


A screenshot of a Mac OS X desktop showing a code editor window. The window title is "watch_stack_example_04.rb — code". The tab bar shows several files: "watch_stack_example_00.rb", "watch_stack_example_02.rb", "watch_stack_example_03.rb", "watch_stack_example_04.rb" (which is the active tab), and "b". The code editor displays the following Ruby code:

```
1 require 'listen'
2
3 def reload!
4   prepare_watcher!
5   load_files
6   mark_constants_as_unloadable!
7 end
8
9 def watch_filepaths(filepaths)
10   listener = Listen.to(*filepaths)
11   listener.change(Proc.new { |mod, add, rem| reload! })
12   listener.start
13 end
14
15 Thread.new { watch_filepaths(watched_filepaths) }
16
```

The status bar at the bottom of the window shows: "Line 15, Column 50; Saved ~/talks/utils-junk-drawer/code/watch_stack_example_04.rb (UTF-8)" on the left, "Spaces: 2" and "Ruby" on the right.

jimweirich/rake



A screenshot of a Mac OS X desktop environment showing a code editor window. The window title is "rake.rb". The code editor displays the following Ruby code:

```
1 class Module
2   def rake_extension(method)
3     if instance_methods.include?(method)
4       $stderr.puts "WARNING: Possible conflict with
5                   Rake extension: #{self}##{method} already
6                   exists"
7     else
8       yield
9     end
10    end
11  end
```

The code is syntax-highlighted: "class" and "def" are blue, "Module" and "method" are black, "if", "instance_methods", "include?", "\$stderr", "puts", "WARNING", "Rake", "extension", "self", and "method" are orange, and the multi-line string is green. Line 7 is highlighted with a gray background.

At the bottom of the window, status bar text includes "Line 7, Column 8", "Spaces: 2", and "Ruby".

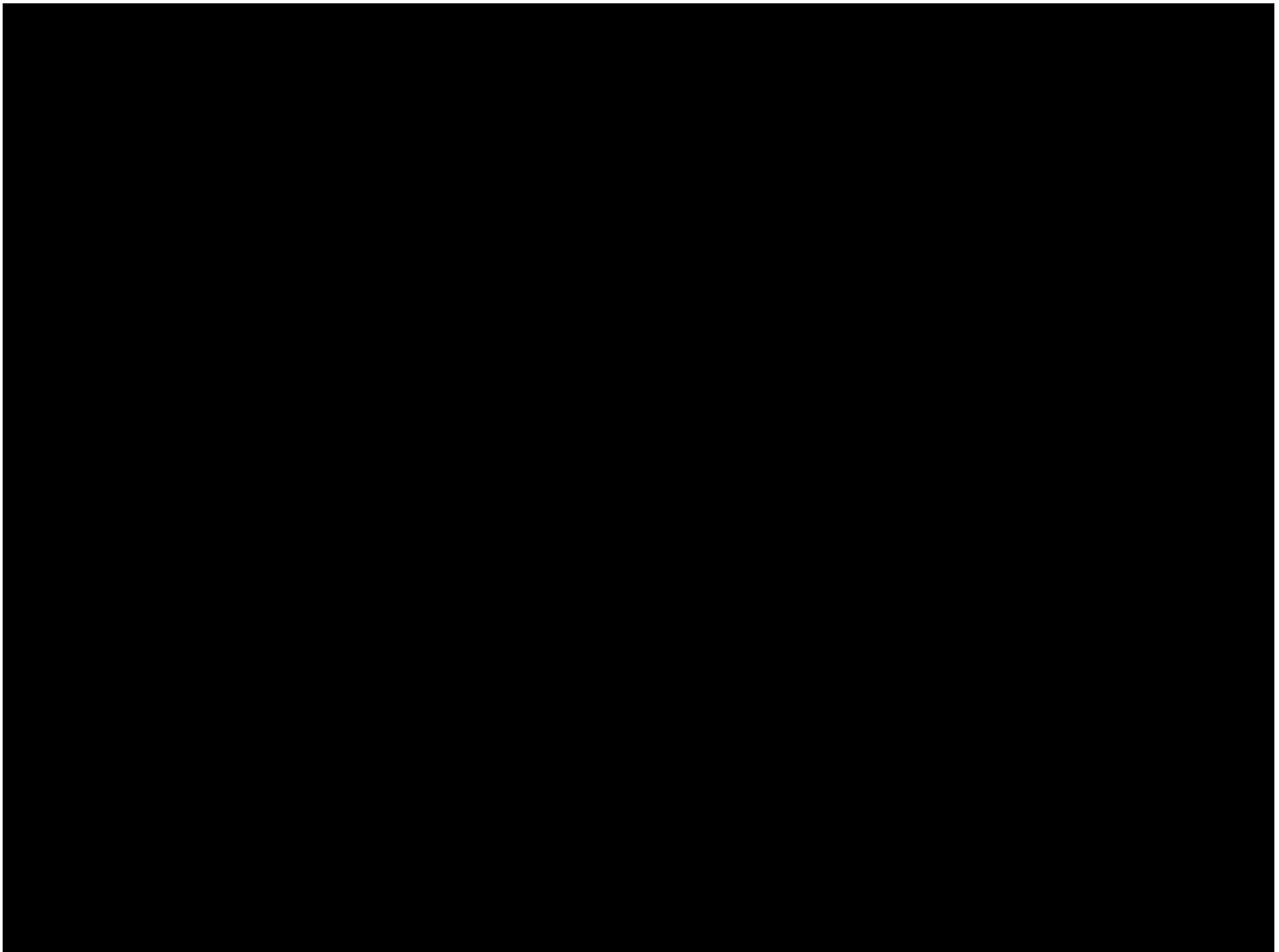
jimweirich/rake



```
rake_example.rb
```

```
1 class String
2   rake_extension("ext") do
3     def ext(newext = '')
4       return self.dup if ['.', '..'].include? self
5       if newext != ''
6         newext = (newext =~ /^\./) ? newext : ("." +
7           newext)
8         dup.sub!(%r(([^\\])\\. [^.\\]*$)) { $1 +
9           newext } || self + newext
10      end
11    end
12  end
```

Line 3, Column 1 Spaces: 2 Ruby

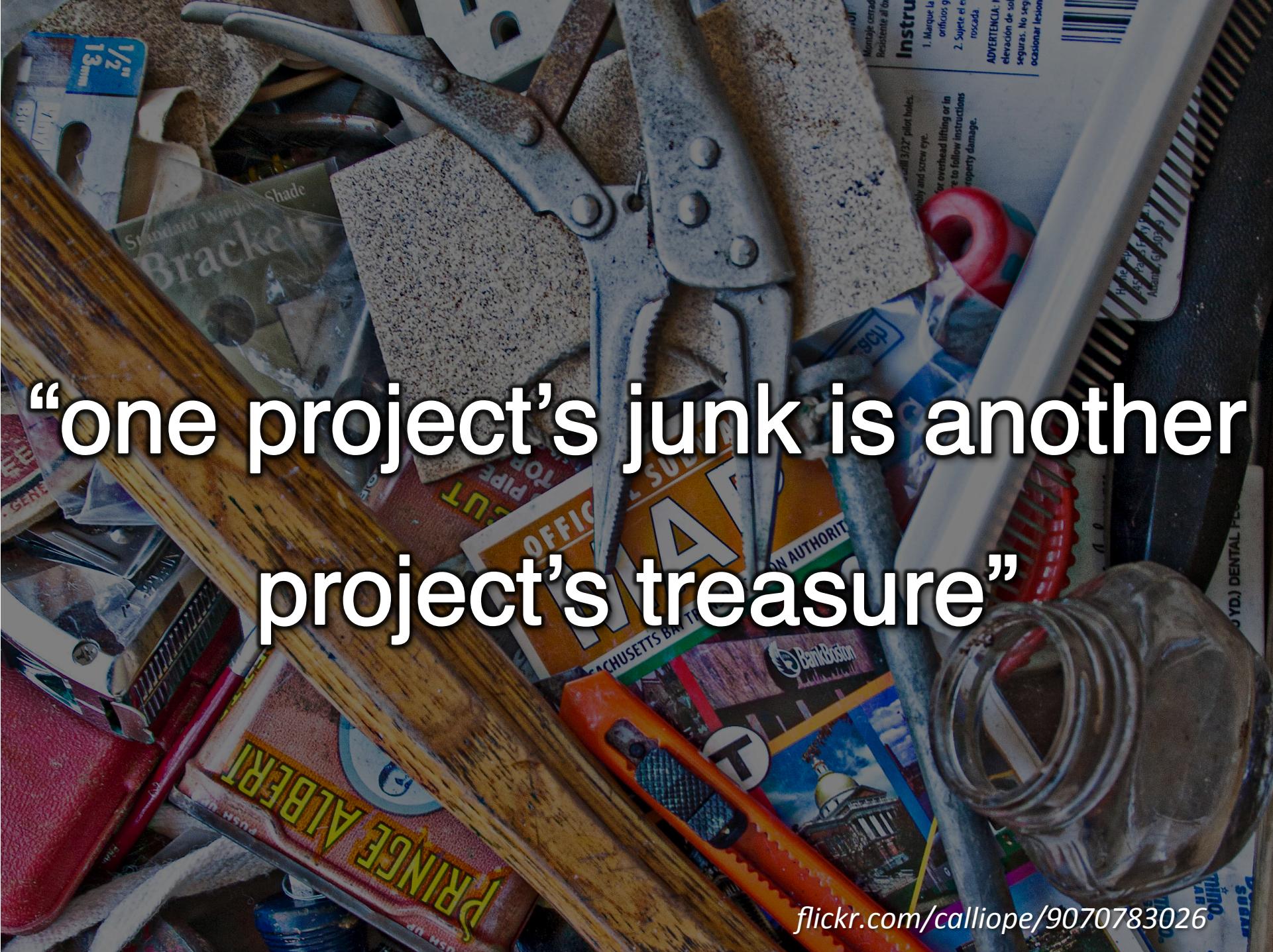


A black and white photograph of a metal door. The door features a series of vertical bars across the top half. Below the bars is a textured, pebbled-surface panel with a keypad consisting of several circular buttons. To the left of the keypad, there is some faint, illegible text. The right side of the door is a smooth, dark panel.

better || worse

flickr.com/19779889@N00/12910103635

Question



“one project’s junk is another
project’s treasure”

flickr.com/calliope/9070783026



JUNK

@franklinwebber

flickr.com/calliope/9070783026