

# How To Make Your Codebase A Joyful Place

ADARSH PANDIT

[twitter.com/adarshp](https://twitter.com/adarshp)

[github.com/adarsh](https://github.com/adarsh)



AND YOU ARE?

# Adarsh Pandit

adarsh

Freelance Ruby Developer



SF



<http://adarsh.io>



Joined on Jul 13, 2011

“I'M NOT A GREAT  
PROGRAMMER; I'M JUST A  
GOOD PROGRAMMER WITH  
GREAT HABITS”

— Kent Beck in “*Refactoring: Improving the Design of Existing Code*”  
Martin Fowler, Kent Beck, John Brant (2012)

# TODAY'S GOALS

1. CONVINCE YOU THAT BEST PRACTICES ARE IMPORTANT FOR YOUR BUSINESS
2. SHARE TACTICS FOR AGREEING ON SOME BEST PRACTICES
3. LIST HOW TO AUTOMATE SOME BEST PRACTICES
4. POLARIZE THE CROWD WITH MY OWN STYLE GUIDE OPINIONS

# TRAILING CONDITIONALS

YES

```
1 class SomeClassName
2   def hello
3     if person.friendly?
4       puts "Hello!"
5     end
6   end
7 end
```

NO

```
1 class SomeClassName
2   def hello
3     puts "Hello!" if person.friendly?
4   end
5 end
```

**“PROGRAMS ARE MEANT  
TO BE READ BY HUMANS  
AND ONLY INCIDENTALLY  
FOR COMPUTERS TO  
EXECUTE.”**

— H. Abelson and G. Sussman  
*(in “Structure and Interpretation of Computer Programs”)*

- 1. CONVINCE YOU THAT BEST PRACTICES ARE IMPORTANT FOR YOUR BUSINESS**
- 2. SHARE TACTICS FOR AGREEING ON SOME BEST PRACTICES**
- 3. LIST HOW TO AUTOMATE SOME BEST PRACTICES**
- 4. POLARIZE THE CROWD WITH MY OWN STYLE GUIDE OPINIONS**

# BEST PRACTICES: WHY DO THEY MATTER?

## BECAUSE MONEY

developer.happy #=> budget.recruiting -= lots

developer.fast #=> developer.roi += lots

bugs.destroy\_all #=> customer.happy

1. CONVINCE YOU THAT BEST PRACTICES ARE IMPORTANT FOR YOUR BUSINESS
2. SHARE TACTICS FOR AGREEING ON SOME BEST PRACTICES
3. LIST HOW TO AUTOMATE SOME BEST PRACTICES
4. POLARIZE THE CROWD WITH MY OWN STYLE GUIDE OPINIONS

# LET'S AGREE AS A TEAM: BEST PRACTICES ARE IMPORTANT BECAUSE

- We want to engender a learning culture of continuous improvement
- We want to be proud of our codebase and excited to go to work each day
- Argue less about dumb stuff
- Routine tasks should be automated

# HOW WILL WE DISCUSS BEST PRACTICES?

## STARTING OUT

- Determine which best practices are valuable for us. It's not all of them.
- Openly talk about why this is hard:  
feeling smart! feeling dumb! history! business!  
individuality! Ruby!
- Draw a picture of where we want to go as an organization
- Get leadership buy-in, which may mean adopting their style (to start)
- Adopt an experimental, "let's try it out!" attitude

# HOW WILL WE DISCUSS BEST PRACTICES?

## ONGOING

- Write it in Markdown, under version control.
- PRs are discussions because we are developers.
- Agree on a democratic approach but have a tiebreaker
- Start with one thing we already agree on
- Follow up with one thing no one agrees on

# KEYWORD ARGUMENTS ALWAYS

YES

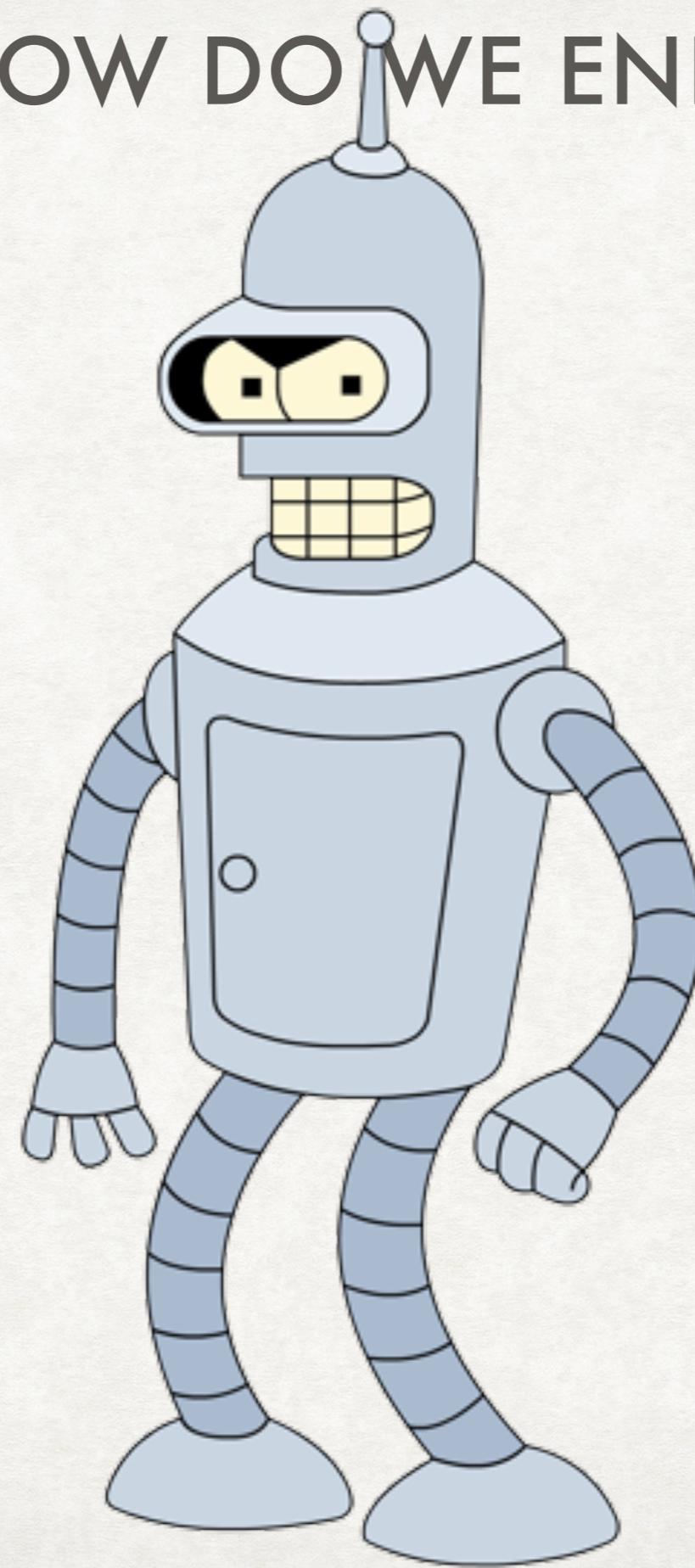
```
1 class SomeClassName
2   def hello(person:)
3     if person.friendly?
4       puts "Hello!"
5   end
6 end
7 end
```

NO

```
1 class SomeClassName
2   def hello(person)
3     if person.friendly?
4       puts "Hello!"
5   end
6 end
7 end
```

1. CONVINCE YOU THAT BEST PRACTICES ARE IMPORTANT FOR YOUR BUSINESS
2. SHARE TACTICS FOR AGREEING ON SOME BEST PRACTICES
3. LIST HOW TO AUTOMATE SOME BEST PRACTICES
4. POLARIZE THE CROWD WITH MY OWN STYLE GUIDE OPINIONS

OK GREAT: HOW DO WE ENFORCE THIS?



59		- @include span-columns(6 of 10);
	59	+ @include span-columns(6 of 10);

 added a note 9 days ago edit ×

I don't understand this change, I think there is an extra space after the semicolon.

 added a note 8 days ago edit ×

Totally true. I guess it was my editor. It is supposed to clean whitespace but maybe I had the cursor there.

```
39
40 + .ct-series-c {
41 +   stroke: $chart-color-green;
42 +
43 +   path {
```



**houndci** added a note 2 days ago



Selector should have depth of applicability no greater than 2, but was 3



██████████ added a note 2 days ago



I really need it to be nested this way.



**adarsh** added a note 2 days ago



Sorry, Hound.



██████████ added a note a day ago



# PREAMBLE: GIT HOOKS

brigade / **overcommit** Watch 75 Unstar 1,311 Fork 94

Code Issues 18 Pull requests 0 Pulse Graphs

A fully configurable and extendable Git hook manager

1,233 commits 2 branches 59 releases 42 contributors

Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

File	Description	Time
 sds	Upgrade RuboCop 0.39.0 -> 0.40.0 for Travis	Latest commit 9636eb3 12 days ago
 bin	Auto-load Bundler context for CLI	2 months ago
 config	Add *.ru matcher to rubocop hook files list	13 days ago
 lib	Upgrade RuboCop 0.39.0 -> 0.40.0 for Travis	12 days ago
 libexec	Gracefully handle .git submodule files	2 years ago
 logo	Add Overcommit logo to README with attribution	4 months ago
 spec	Upgrade RuboCop 0.39.0 -> 0.40.0 for Travis	12 days ago
 template-dir/hooks	Add global quiet option for silencing hook runs	2 months ago
 .editorconfig	Add .editorconfig file	2 years ago
 .gitignore	Add code coverage integration with Coveralls	a year ago

# PREAMBLE: GIT HOOKS

```
~/my-repo git:master >>> |
```

+

# PREAMBLE: GIT HOOKS

www.guoxiang.me/posts/28-how-to-write-a-custom-overcommit-precommit-git-hook-in-4-steps



## Guo Xiang

I built [RubyBench.org](#)

# How to Write a Custom Overcommit PreCommit Git Hook in 4 Steps

By Guo Xiang on 16 AUG 2014

*This post summarizes the first ever Ruby talk I gave at a [RubySG](#) meetup*

*\*Follow the [instructions here](#) to install Overcommit before reading the tutorial\**

Despite [Overcommit](#) being an opinionated git hook manager, it is fully configurable and extendable as well. In this tutorial, I will walk you through on how you can write a pre-commit hook (with tests) that enforces a maximum length for each line in your files through the use of the APIs provided by the Overcommit gem.

## Step 1. Putting the files in the right directory

By default, Overcommit looks for your custom hooks in the `git_hooks` folder which we can configure

# PRIVATE ATTR\_READERS

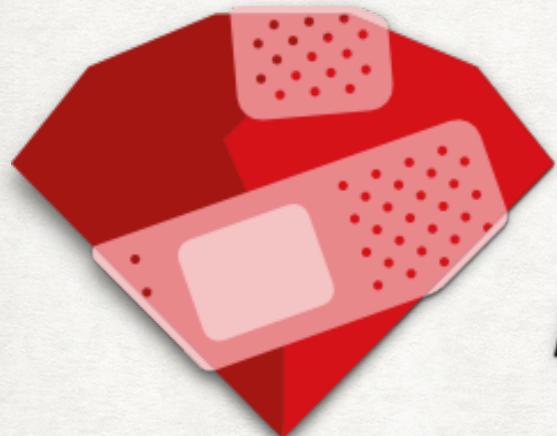
YES

```
1 class SomeClassName
2   def initialize(person:)
3     @person = person
4   end
5
6   def hello
7     puts "Hello, #{person.name}!"
8   end
9
10 private
11
12 attr_reader :person
13 end
```

NO

```
1 class SomeClassName
2   def initialize(person:)
3     @person = person
4   end
5
6   def hello
7     puts "Hello, #{@person.name}!"
8   end
9 end
```

# CODE QUALITY OPEN SOURCE TOOLS



# Reek

seattlerb / flay

seattlerb / flog

square / cane

Code Issues 6 Pull requests 0 Pulse

Code quality threshold checking as part of your build

## SandiMeter

build passing gem version 1.2.0

Static analysis tool for checking your Ruby code for Sandi Metz' four rules.

- 100 lines per class
- 5 lines per method
- 4 params per method call (and don't even try cheating with hash params)
- 1 instance variable per controller action

# CODE QUALITY OPEN SOURCE TOOLS

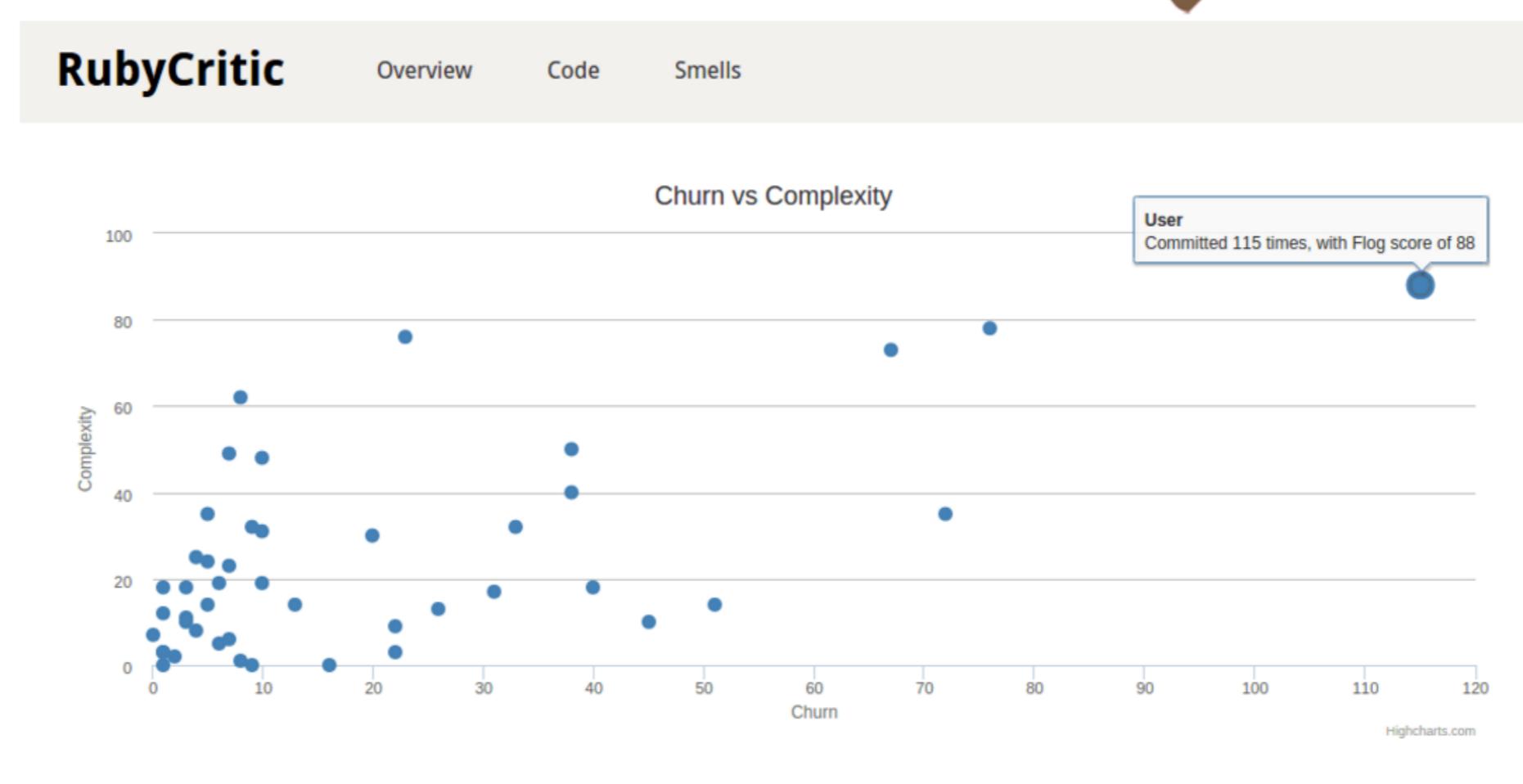
## RubyCritic

gem version 2.9.1 build passing code climate 4.0

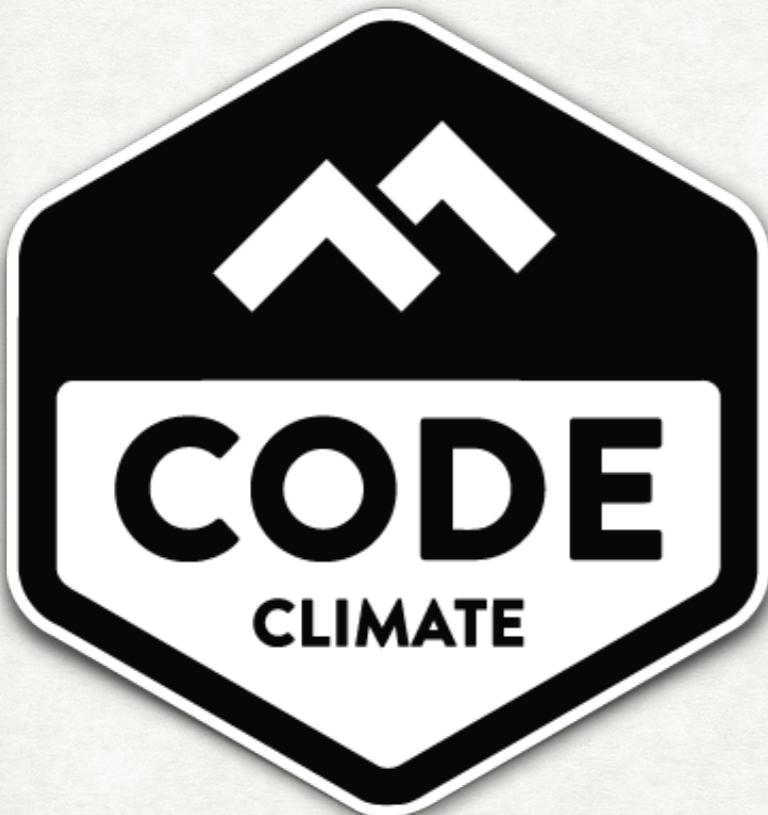
RubyCritic is a gem that wraps around static analysis gems such as [Reek](#), [Flay](#) and [Flog](#) to provide a quality report of your Ruby code.

This gem provides features such as:

1. An overview of your project:

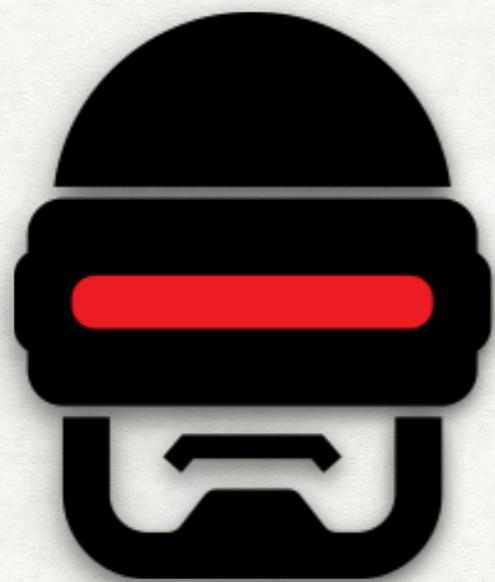


# CODE QUALITY PAID SERVICES



SensioLabsInsight

# CODE STYLE OPEN SOURCE TOOLS



# RuboCop

 [bbatsov / ruby-style-guide](#)

[!\[\]\(7f027cf06456dca531ef477a54774593\_img.jpg\) Code](#) [!\[\]\(66de7ffc73865763bee241dc1d2c4631\_img.jpg\) Issues 64](#) [!\[\]\(cb0d25c14dc3cbe7e4be93d8ed1c94e3\_img.jpg\) Pull requests 37](#)

A community-driven Ruby coding style guide

# CODE STYLE

## PAID SERVICES

KEEP IN MIND THAT I'M SELF-TAUGHT, SO MY CODE MAY BE A LITTLE MESSY.

LEMMIE SEE-  
I'M SURE  
IT'S FINE.



...WOW.  
THIS IS LIKE BEING IN A HOUSE BUILT BY A CHILD USING NOTHING BUT A HATCHET AND A PICTURE OF A HOUSE.

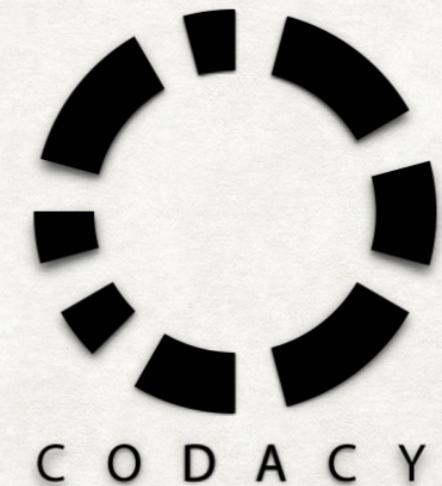


IT'S LIKE A SALAD RECIPE WRITTEN BY A CORPORATE LAWYER USING A PHONE AUTOCORRECT THAT ONLY KNEW EXCEL FORMULAS.



IT'S LIKE SOMEONE TOOK A TRANSCRIPT OF A COUPLE ARGUING AT IKEA AND MADE RANDOM EDITS UNTIL IT COMPILED WITHOUT ERRORS.

OKAY, I'LL READ A STYLE GUIDE.



# TEST COVERAGE OPEN SOURCE TOOLS

 [colszowka / simplecov](#) Watch ▾ 63 Star 2,622 Fork 262

[Code](#) [Issues 48](#) [Pull requests 12](#) [Pulse](#) [Graphs](#)

Code coverage for Ruby 1.9+ with a powerful configuration library and automatic merging of coverage across test suites <https://www.ruby-toolbox.com/projects/simplecov>

# TEST COVERAGE

## PAID SERVICES



Travis CI

The logo for CircleCI consists of a teal circular icon with a white 'C' inside, followed by the word 'circleci' in a lowercase sans-serif font.



Jenkins

# SECURITY PATCHES



[civisanalytics / ruby\\_audit](#)

[Code](#) [Issues 0](#) [Pull requests 0](#) [Wiki](#) [Pulse](#)

Checks Ruby and RubyGems against known security vulnerabilities.

[rubysec / bundler-audit](#)

[Code](#) [Issues 11](#) [Pull req](#)

Patch-level verification for Bundler

# SECURITY PATCHES

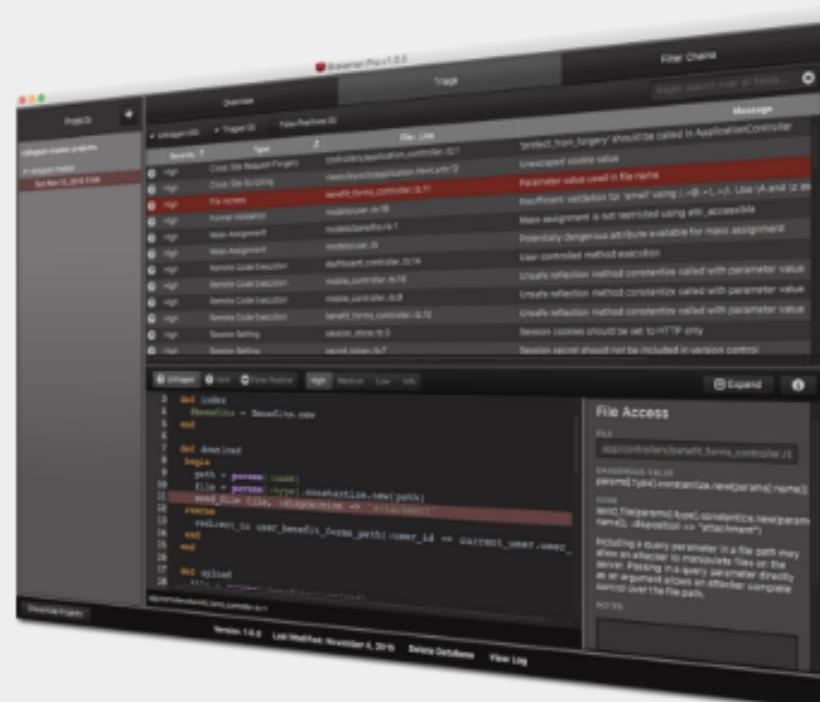
## PAID SERVICES

# Brakeman Pro

**Brakeman Pro Version 1.2.1 is here!**

Brakeman Pro is a static analysis security tool (SAST) for Ruby on Rails applications. It searches for potential security vulnerabilities by scanning the source code of Rails applications.

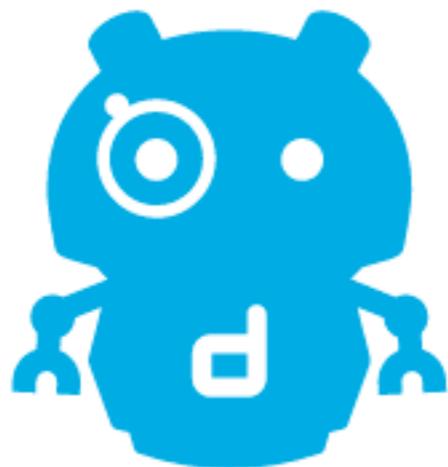
Brakeman Pro provides an interface for reviewing, managing and reporting on warnings across multiple scans and applications.



# Appcanary



# UPDATING YOUR GEMS



depbot

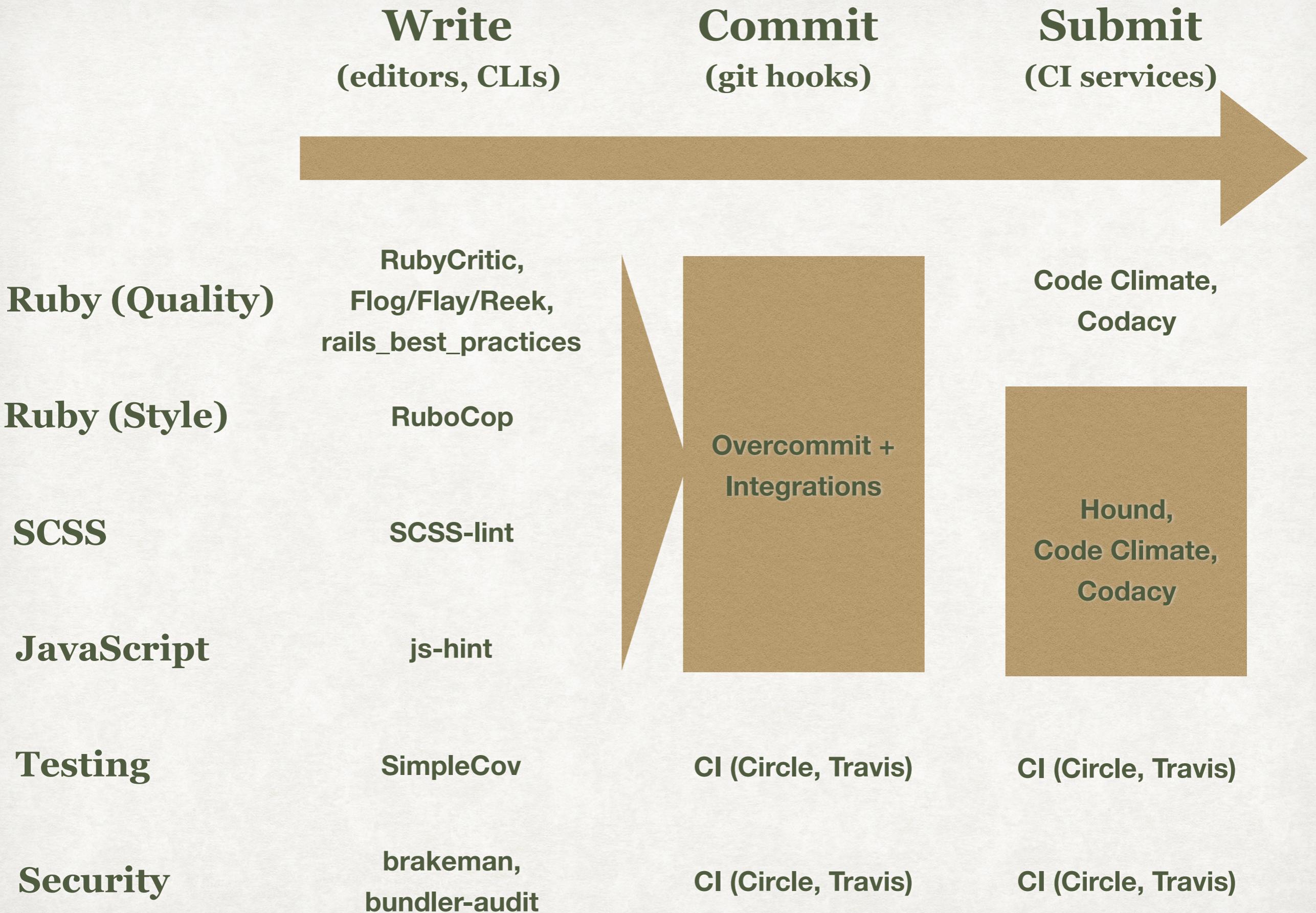
← Adarsh Pandit →

Software Developer

Save Money and Be Happier by Updating Your Gems Every  
Monday Morning

Nov 16, 2015 By Adarsh Pandit in writing

As a newcomer to Rails, understanding Gems and the Gemfile is fairly straightforward: You include a gem that you need and you get it included everywhere.



THANKS!



# APPENDIX

# JOKEY ANTI PATTERN

## THIS JUST ISN'T IDIOMATIC

