☆Be a Good Ruby Citizen☆

Use and contribute code like you give a damn.

“WHY?”

"We've got to have rules and obey them. After all, we're not savages."

Dr Nic's 8 Steps

http://bit.ly/othercode

Dr Nic's 8 Steps

Dr Nic's 8 Steps

1. Get annoyed by a defect or missing feature

Dr Nic's 8 Steps

1. Get annoyed by a defect or missing feature
2. Find the source

Dr Nic's 8 Steps

1. Get annoyed by a defect or missing feature
2. Find the source
3. Checkout the source

Dr Nic's 8 Steps

1. Get annoyed by a defect or missing feature
2. Find the source
3. Checkout the source
4. Snoop around

Dr Nic's 8 Steps

1. Get annoyed by a defect or missing feature
2. Find the source
3. Checkout the source
4. Snoop around
5. Make changes

Dr Nic's 8 Steps

1. Get annoyed by a defect or missing feature
2. Find the source
3. Checkout the source
4. Snoop around
5. Make changes
6. Refresh the code from the repository (svn up)

Dr Nic's 8 Steps

1. Get annoyed by a defect or missing feature
2. Find the source
3. Checkout the source
4. Snoop around
5. Make changes
6. Refresh the code from the repository (svn up)
7. Create the patch

Dr Nic's 8 Steps

1. Get annoyed by a defect or missing feature
2. Find the source
3. Checkout the source
4. Snoop around
5. Make changes
6. Refresh the code from the repository (svn up)
7. Create the patch
8. Submit patch

Let’s simplify

5 Steps

5 Steps

1. Get annoyed by a defect or missing feature

5 Steps

1. Get annoyed by a defect or missing feature
2. Find & checkout the source

5 Steps

1. Get annoyed by a defect or missing feature
2. Find & checkout the source
3. Create a branch & write a failing test

5 Steps

1. Get annoyed by a defect or missing feature
2. Find & checkout the source
3. Create a branch & write a failing test
4. Write code to pass your test & commit

5 Steps

1. Get annoyed by a defect or missing feature
2. Find & checkout the source
3. Create a branch & write a failing test
4. Write code to pass your test & commit
5. Publish your code & submit your patch

OK, 6 Steps

1. Get annoyed by a defect or missing feature
2. Find & checkout the source
3. Create a branch & write a failing test
4. Write code to pass your test & commit
5. Publish your code & submit your patch
6. Pray for acceptance...

You’re done!

You’re done?

MINASWAN

☆Be a Good Ruby Citizen☆

Use and contribute code like you give a damn.

Own Your Code

Own Your Code

- Get familiar
- Take responsibility
- Don't ask for permission
- Take care

If you use it, you own it

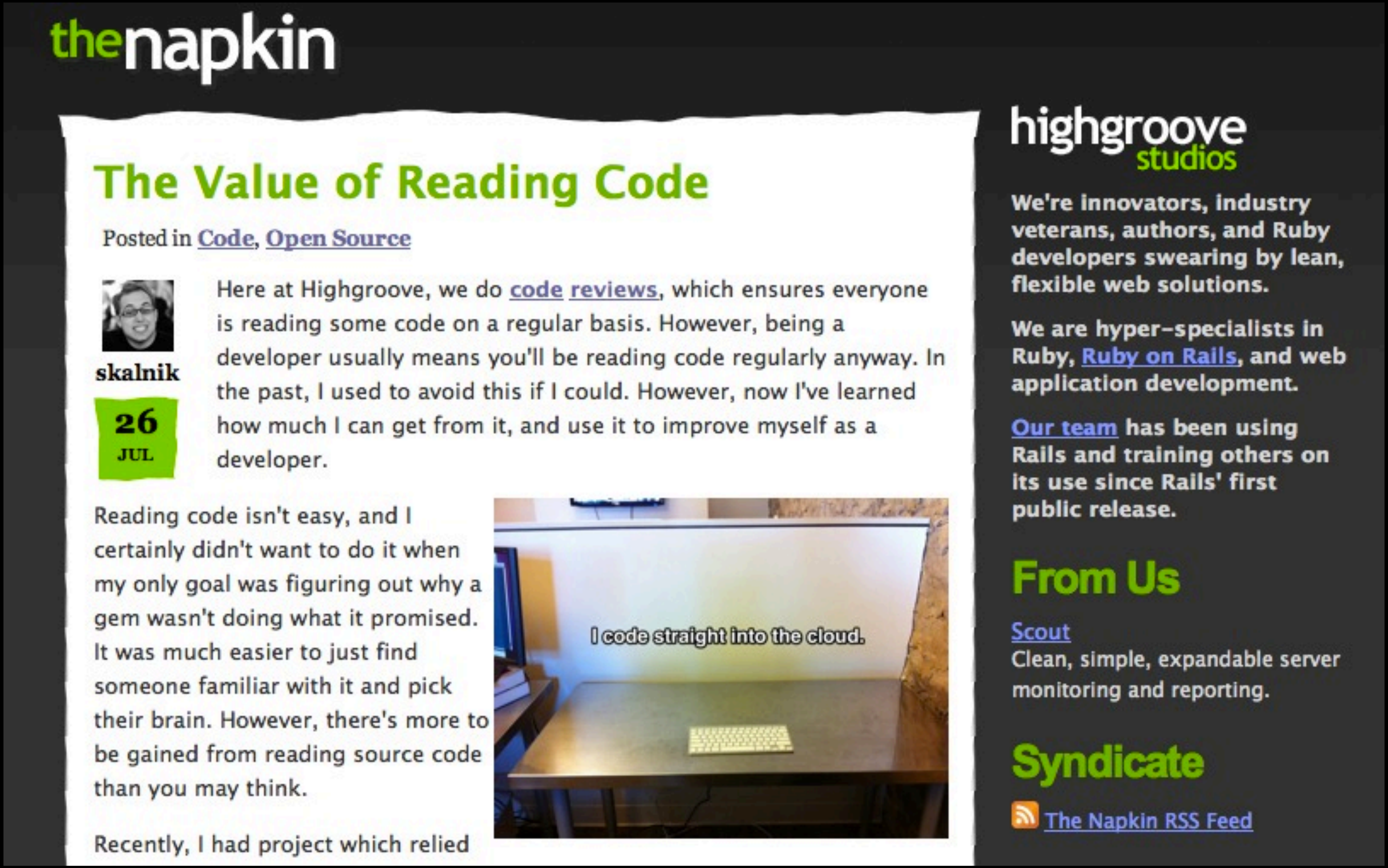
Owners will
take responsibility for and
take care of their possessions

First, get familiar

Read

Mike Skalnik

http://bit.ly/reading-code



thenapkin

The Value of Reading Code

Posted in Code, Open Source

skalnik

26 JUL

Here at Highgroove, we do code reviews, which ensures everyone is reading some code on a regular basis. However, being a developer usually means you'll be reading code regularly anyway. In the past, I used to avoid this if I could. However, now I've learned how much I can get from it, and use it to improve myself as a developer.

Reading code isn't easy, and I certainly didn't want to do it when my only goal was figuring out why a gem wasn't doing what it promised. It was much easier to just find someone familiar with it and pick their brain. However, there's more to be gained from reading source code than you may think.

Recently, I had project which relied

highgroove studios

We're innovators, industry veterans, authors, and Ruby developers swearing by lean, flexible web solutions.

We are hyper-specialists in Ruby, Ruby on Rails, and web application development.

Our team has been using Rails and training others on its use since Rails' first public release.

From Us

Scout
Clean, simple, expandable server monitoring and reporting.

Syndicate

The Napkin RSS Feed

“Now that I've grown used to reading code, I look at it as an opportunity to look at someone else's programming style, and hopefully learn a thing or two.”

Learning happens slowly.

If you haven’t already,
begin now.

Understand

Now, you know
So take responsibility

Use your knowledge

acts_as_tree

Get annoyed by a defect or missing feature

acts_as_tree is not a gem (circa 2009)

Find & checkout the source

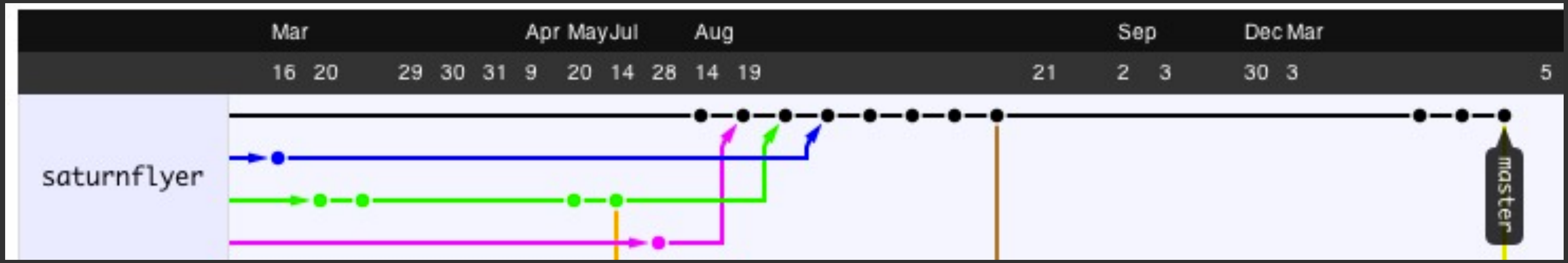
Github to the rescue

Create a branch & write a failing test

hmm... I skipped that.

Write code to pass your test & commit

more like steal other's code and commit



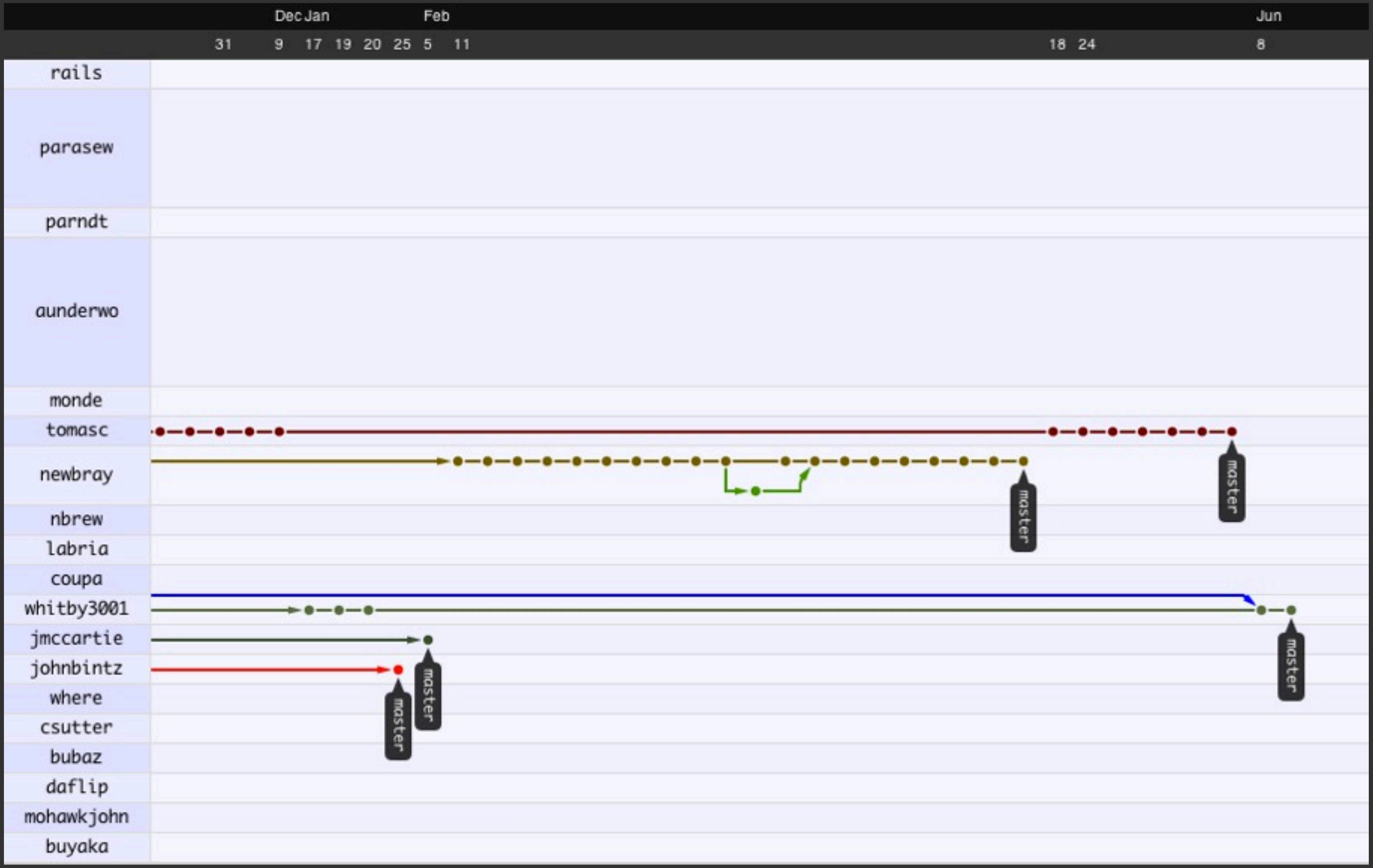
Publish your code & submit your patch

Didn’t bother with the patch

Accept the Thank-yous

Github built the gem

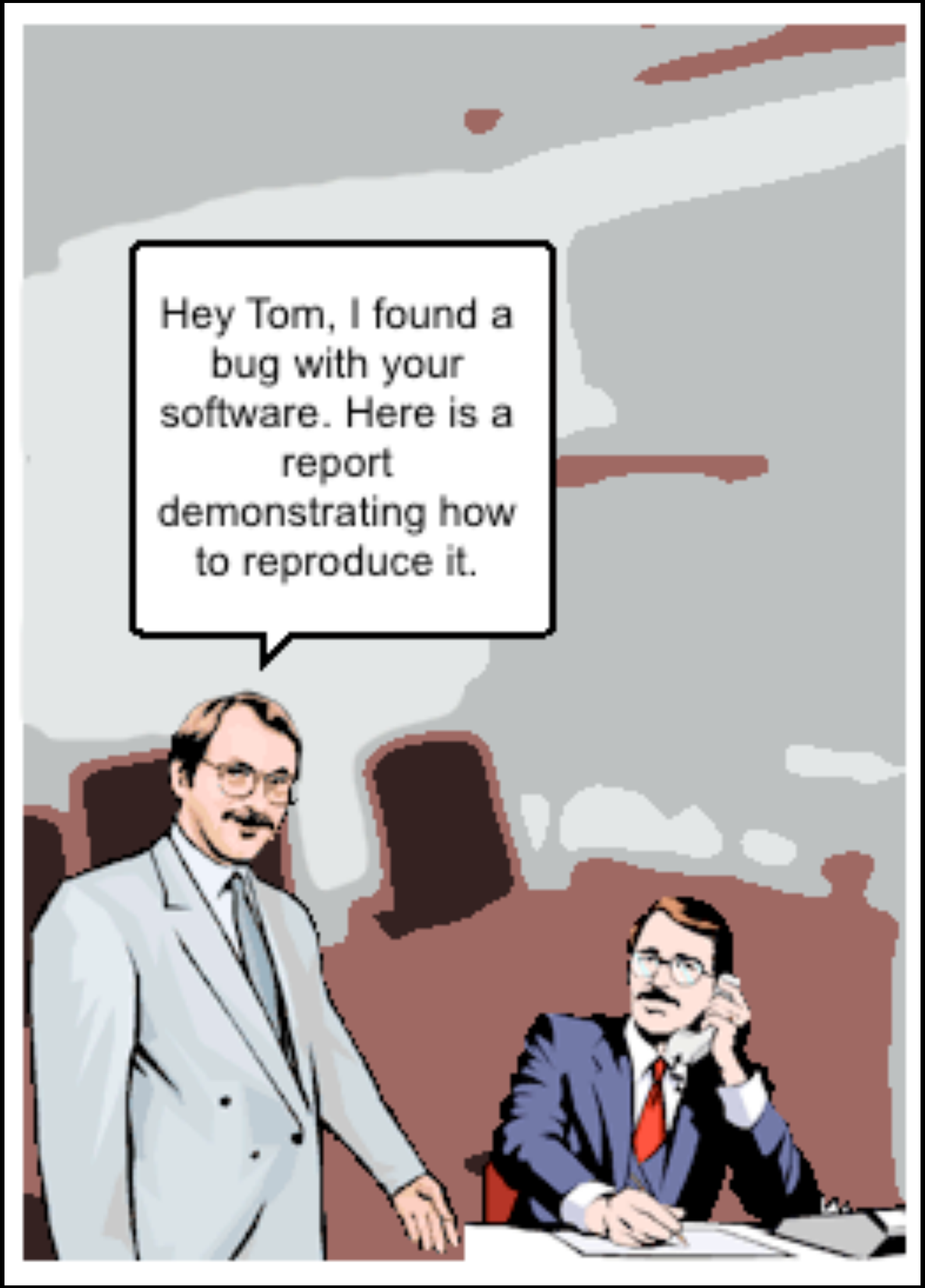
It’s ripe again!



Don't ask for permission

Brian Cardarella

http://bit.ly/patches-welcome


Hey Tom, I found a bug with your software. Here is a report demonstrating how to reproduce it.

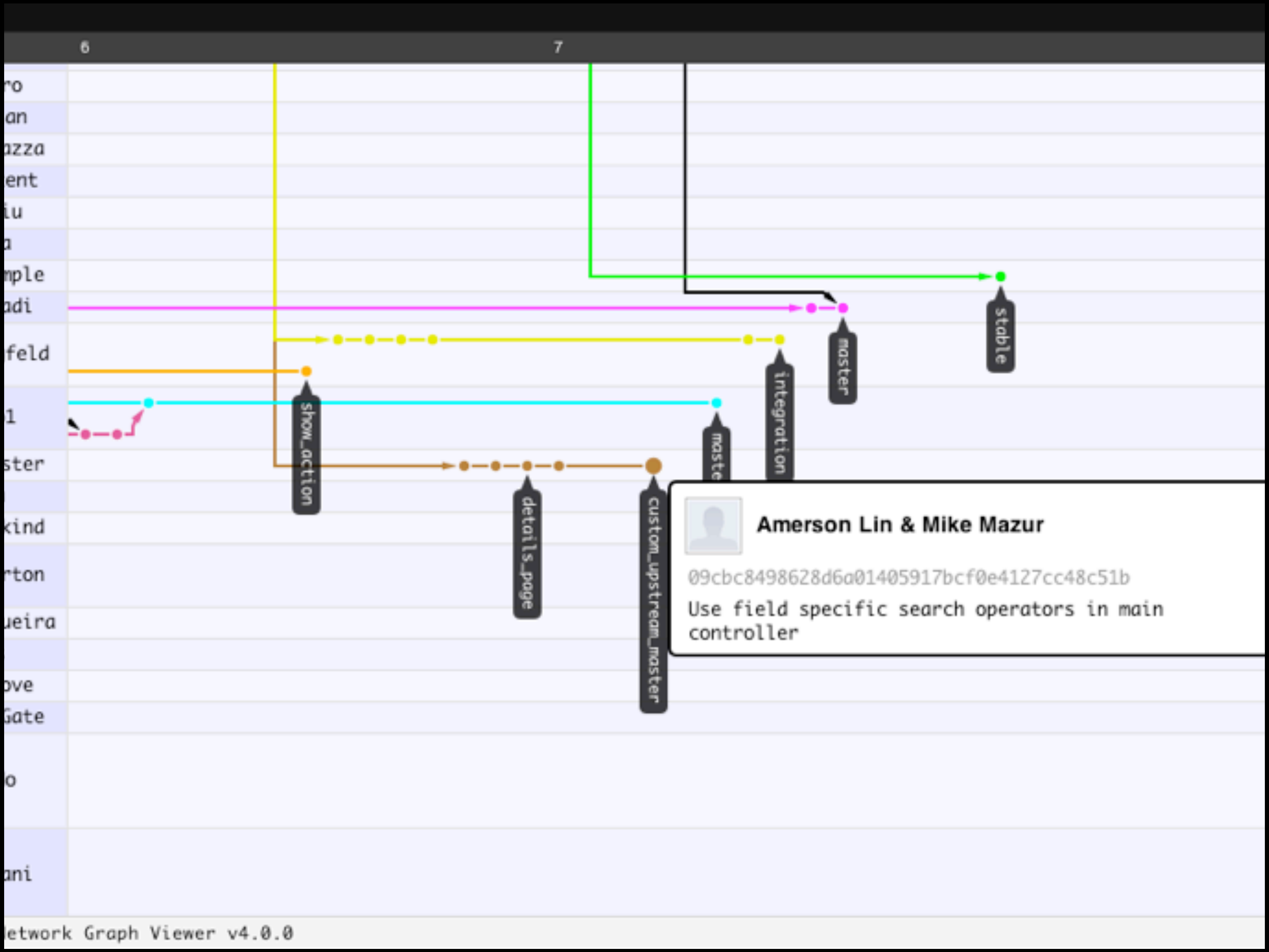
Patches welcome.

Did you just tell me to go fuck myself?
I believe I did, Bob.

Be a leader

Curate other’s commits

Steal other’s commits



Amerson Lin & Mike Mazur

09cbc8498628d6a01405917bcf0e4127cc48c51b

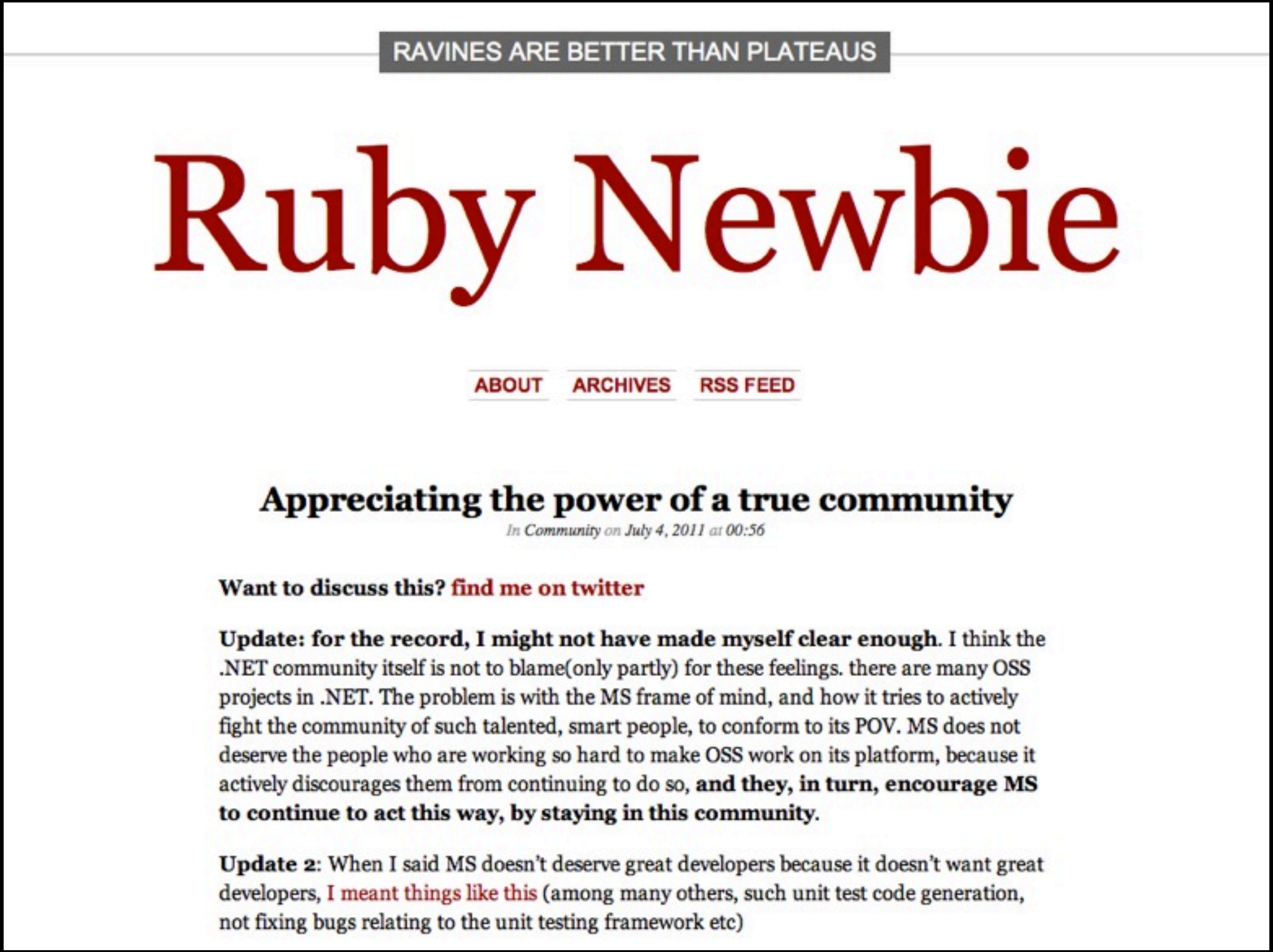
Use field specific search operators in main controller

Your best friend

“How do I take care?”

Roy Osherove

http://bit.ly/true-community



RAVINES ARE BETTER THAN PLATEAUS

Ruby Newbie

ABOUT ARCHIVES RSS FEED

Appreciating the power of a true community

In Community on July 4, 2011 at 00:56

Want to discuss this? find me on twitter

Update: for the record, I might not have made myself clear enough. I think the .NET community itself is not to blame(only partly) for these feelings. there are many OSS projects in .NET. The problem is with the MS frame of mind, and how it tries to actively fight the community of such talented, smart people, to conform to its POV. MS does not deserve the people who are working so hard to make OSS work on its platform, because it actively discourages them from continuing to do so, **and they, in turn, encourage MS to continue to act this way, by staying in this community.**

Update 2: When I said MS doesn't deserve great developers because it doesn't want great developers, I meant things like this (among many others, such unit test code generation, not fixing bugs relating to the unit testing framework etc)

Discuss bugs with others

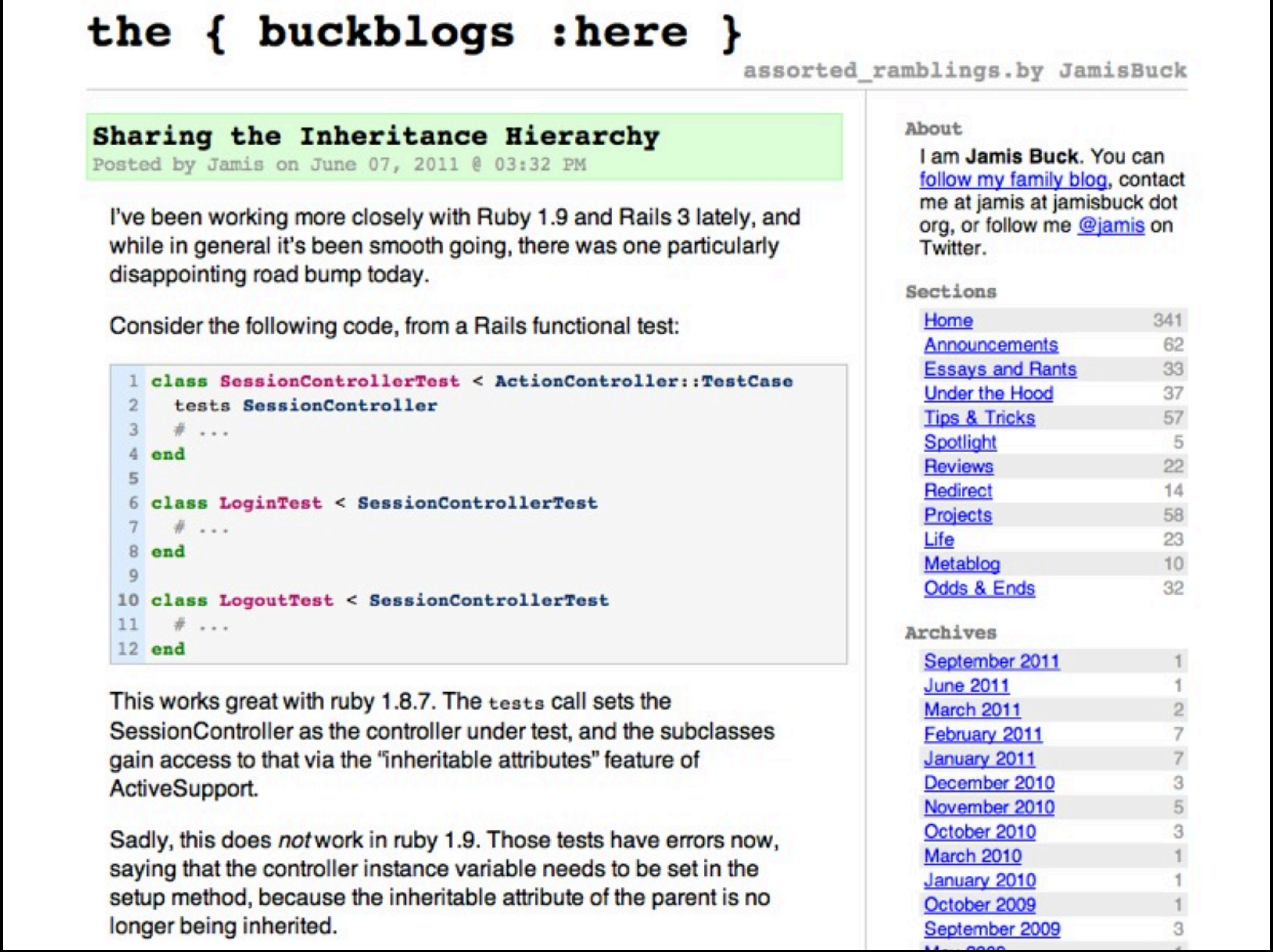
Discuss goals &
direction with others

Advocate for mutual benefit

Read and respond
to mailing lists and blogs

Jamis Buck

http://bit.ly/mini-frustration



the { buckblogs :here }

assorted_ramblings.by JamisBuck

Sharing the Inheritance Hierarchy

Posted by Jamis on June 07, 2011 @ 03:32 PM

I've been working more closely with Ruby 1.9 and Rails 3 lately, and while in general it's been smooth going, there was one particularly disappointing road bump today.

Consider the following code, from a Rails functional test:

```ruby
class SessionControllerTest < ActionController::TestCase
  tests SessionController
  # ...
end

class LoginTest < SessionControllerTest
  # ...
end

class LogoutTest < SessionControllerTest
  # ...
end
```

This works great with ruby 1.8.7. The `tests` call sets the SessionController as the controller under test, and the subclasses gain access to that via the "inheritable attributes" feature of ActiveSupport.

Sadly, this does *not* work in ruby 1.9. Those tests have errors now, saying that the controller instance variable needs to be set in the setup method, because the inheritable attribute of the parent is no longer being inherited.

About

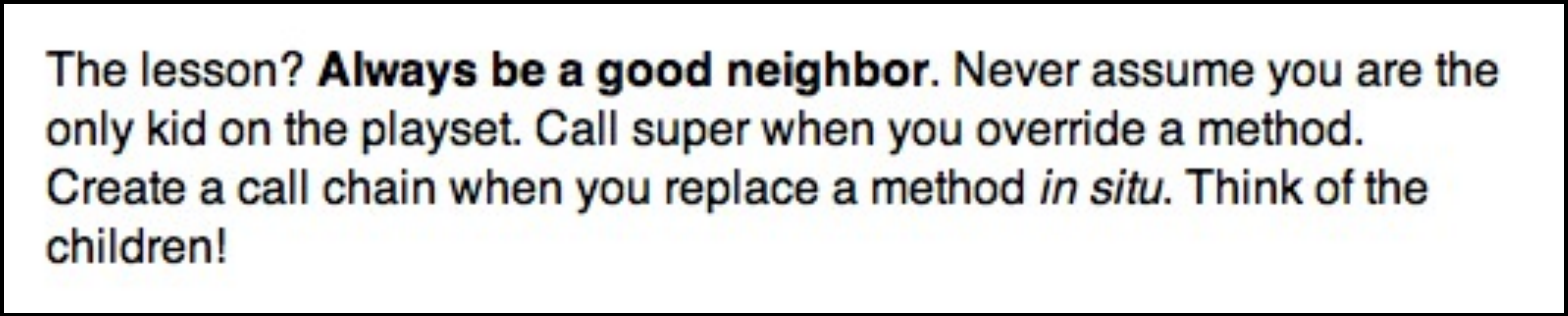
I am **Jamis Buck**. You can follow my family blog, contact me at jamis at jamisbuck dot org, or follow me @jamis on Twitter.

Sections

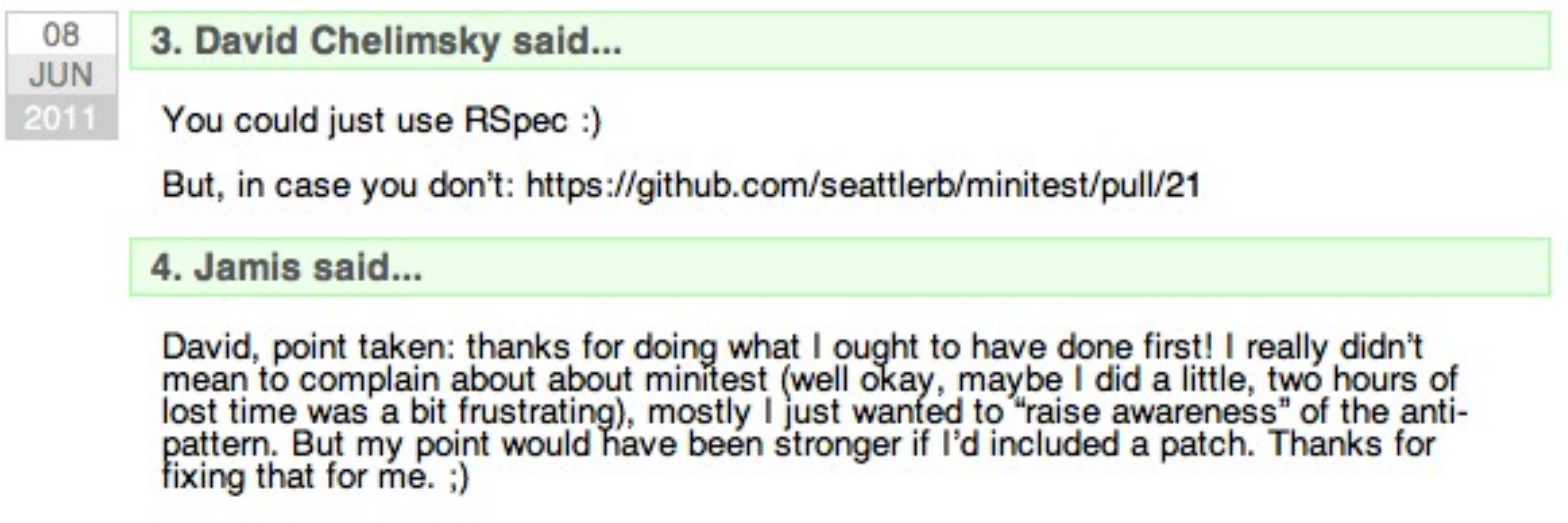
Home	341
Announcements	62
Essays and Rants	33
Under the Hood	37
Tips & Tricks	57
Spotlight	5
Reviews	22
Redirect	14
Projects	58
Life	23
Metablog	10
Odds & Ends	32

Archives

September 2011	1
June 2011	1
March 2011	2
February 2011	7
January 2011	7
December 2010	3
November 2010	5
October 2010	3
March 2010	1
January 2010	1
October 2009	1
September 2009	3



The lesson? **Always be a good neighbor**. Never assume you are the only kid on the playset. Call super when you override a method. Create a call chain when you replace a method *in situ*. Think of the children!



08
JUN
2011

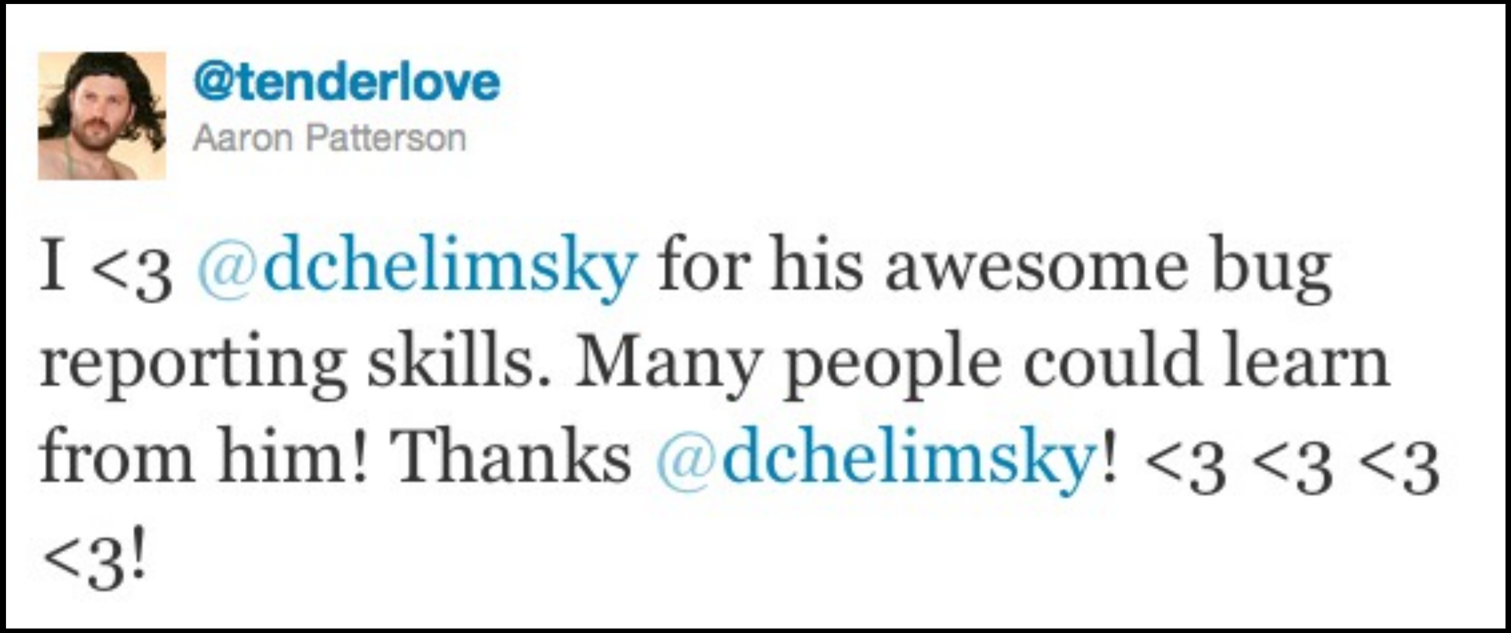
3. David Chelimsky said...

You could just use RSpec :)

But, in case you don't: https://github.com/seattlerb/minitest/pull/21

4. Jamis said...

David, point taken: thanks for doing what I ought to have done first! I really didn't mean to complain about about minitest (well okay, maybe I did a little, two hours of lost time was a bit frustrating), mostly I just wanted to "raise awareness" of the anti-pattern. But my point would have been stronger if I'd included a patch. Thanks for fixing that for me. ;)



@tenderlove
Aaron Patterson

I <3 @dchelimsky for his awesome bug reporting skills. Many people could learn from him! Thanks @dchelimsky! <3 <3 <3 <3!

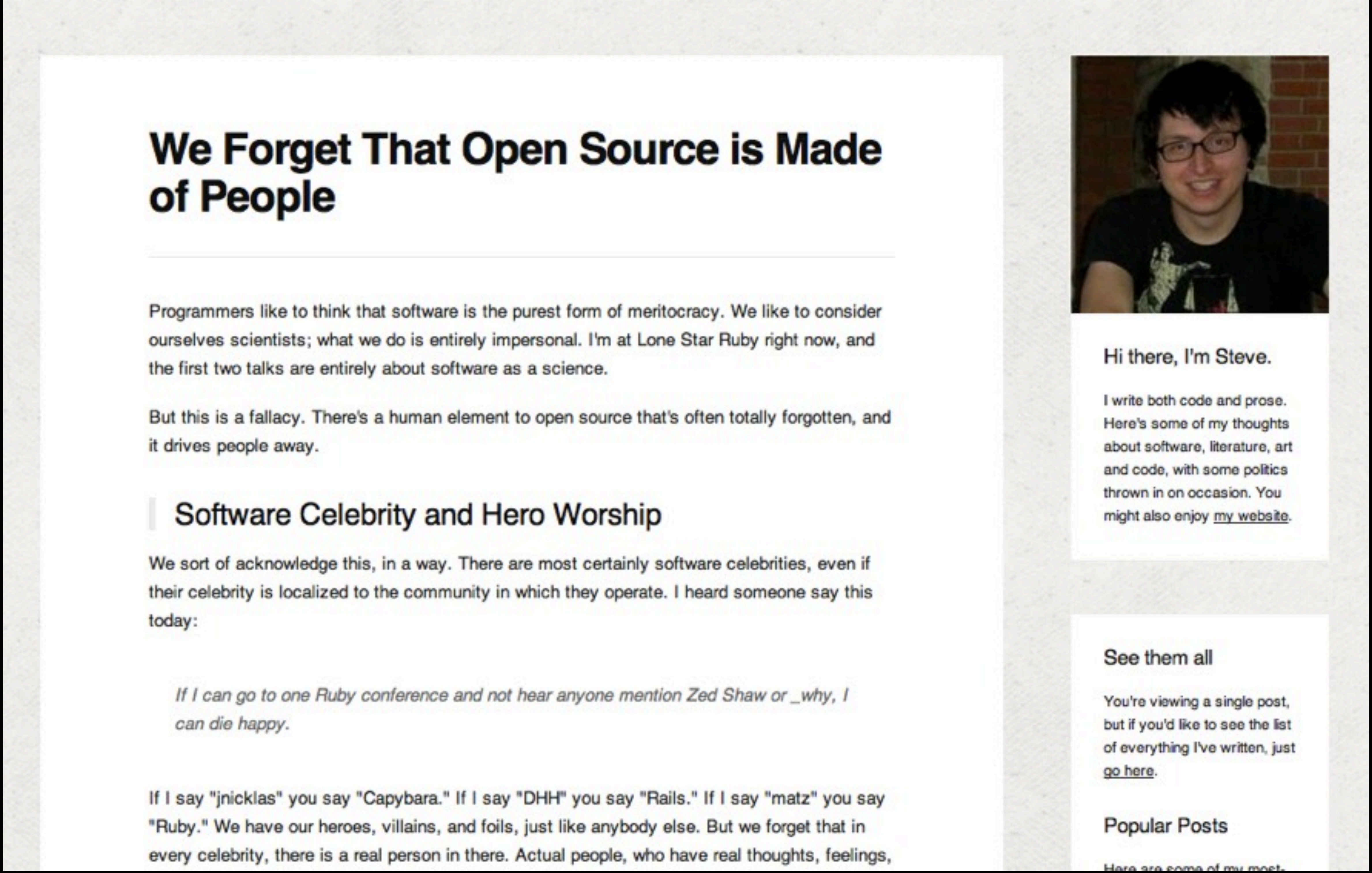
https://twitter.com/#!/
tenderlove/status/
101365041177165826

Remember the newbies!!!

Teach!

Steve Klabnik

http://bit.ly/people-oss



We Forget That Open Source is Made of People

Programmers like to think that software is the purest form of meritocracy. We like to consider ourselves scientists; what we do is entirely impersonal. I'm at Lone Star Ruby right now, and the first two talks are entirely about software as a science.

But this is a fallacy. There's a human element to open source that's often totally forgotten, and it drives people away.

Software Celebrity and Hero Worship

We sort of acknowledge this, in a way. There are most certainly software celebrities, even if their celebrity is localized to the community in which they operate. I heard someone say this today:

> *If I can go to one Ruby conference and not hear anyone mention Zed Shaw or _why, I can die happy.*

If I say "jnicklas" you say "Capybara." If I say "DHH" you say "Rails." If I say "matz" you say "Ruby." We have our heroes, villains, and foils, just like anybody else. But we forget that in every celebrity, there is a real person in there. Actual people, who have real thoughts, feelings,

Hi there, I'm Steve.

I write both code and prose. Here's some of my thoughts about software, literature, art and code, with some politics thrown in on occasion. You might also enjoy my website.

See them all

You're viewing a single post, but if you'd like to see the list of everything I've written, just go here.

Popular Posts

Here are some of my most-

Own Your Code

Own Your Code

- Get familiar
- Take responsibility
- Don't ask for permission
- Take care

Spread the ❤

Use and contribute code like you give a damn

☆Be a Good Ruby Citizen☆

github.com/saturnflyer
@saturnflyer