

# *Continuous Delivery in Ruby*

---

Srushti Ambekallu  
[@srshti](https://twitter.com/srshti)  
[github.com/srushti](https://github.com/srushti)



Brian Guthrie  
[@bguthrie](https://twitter.com/bguthrie)  
[github.com/bguthrie](https://github.com/bguthrie)

**ThoughtWorks®**

Delivering great software



© 2004 Blizzard Entertainment. All rights reserved.

Continuous Integration

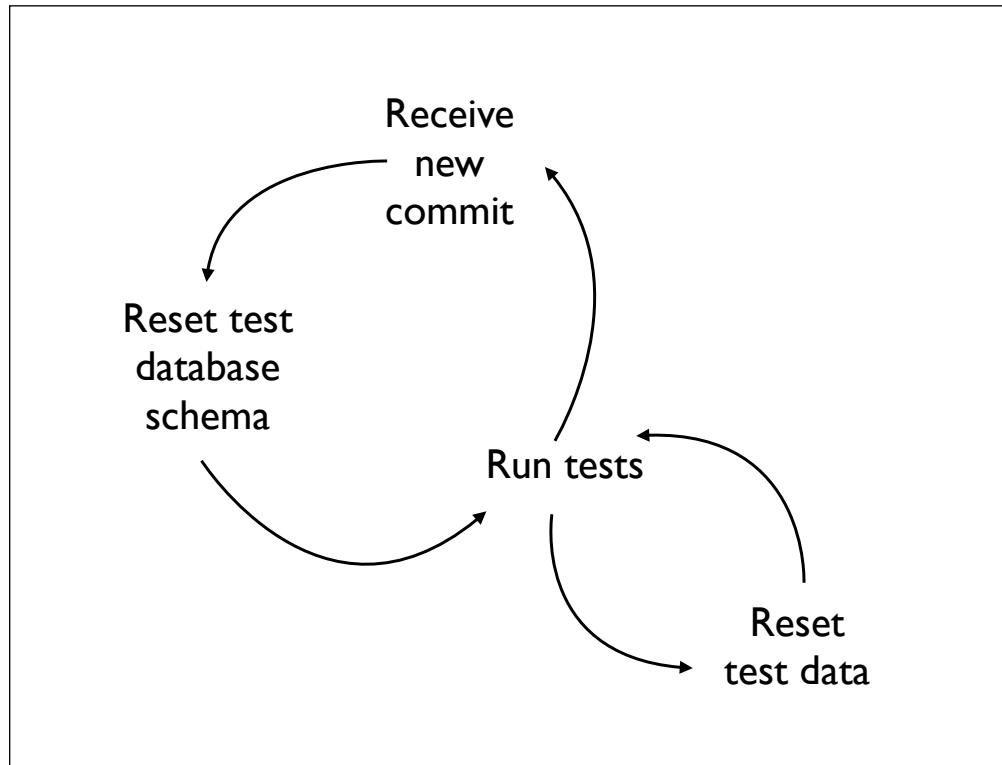
Brian – skeptic (“what do I need tests for?”)

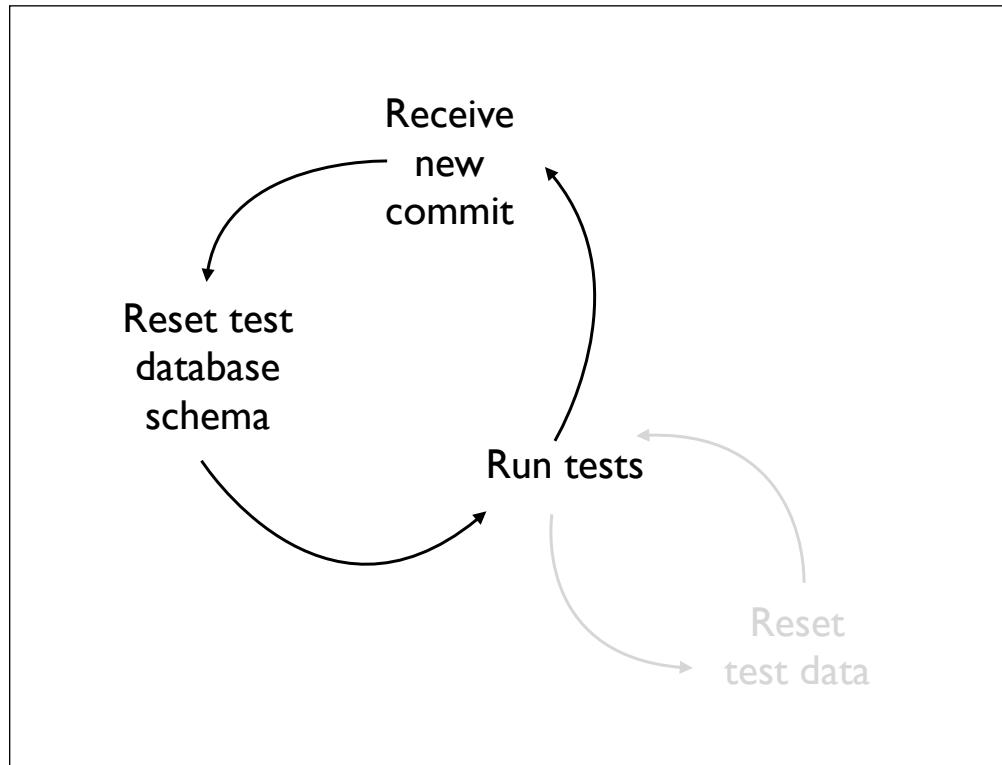
Srushti – champion

## I.Unit and integration tests

RSpec or Test::Unit

Tests of specific units of code  
Tests of multiple layers  
Tests of external integration points



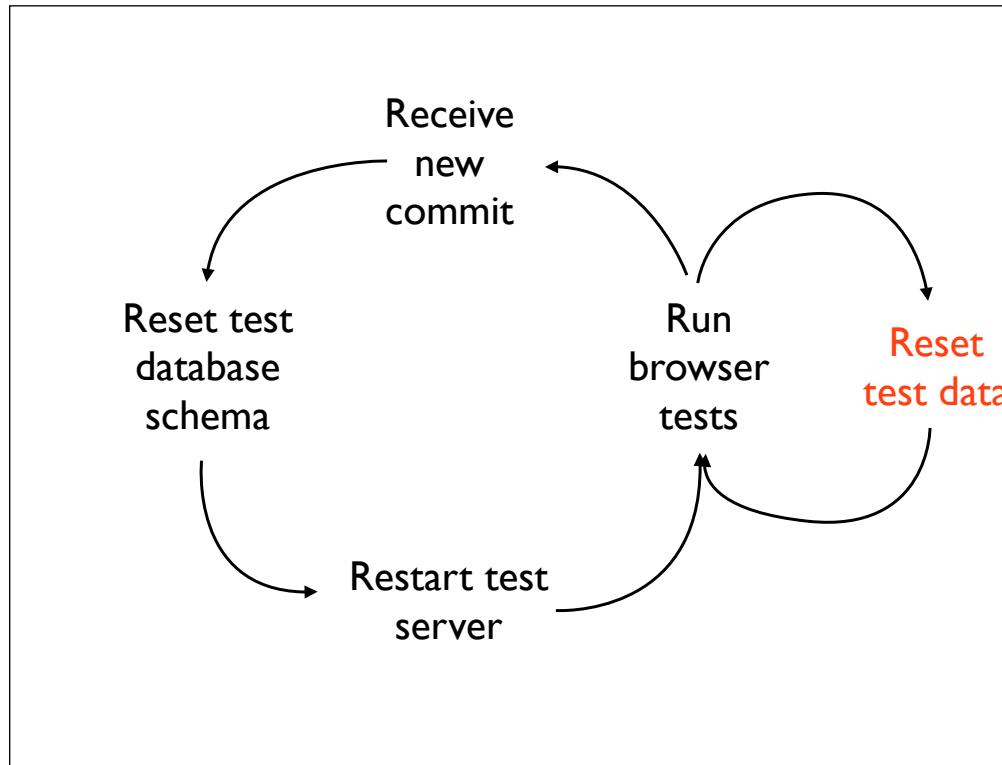


I.Unit and integration tests

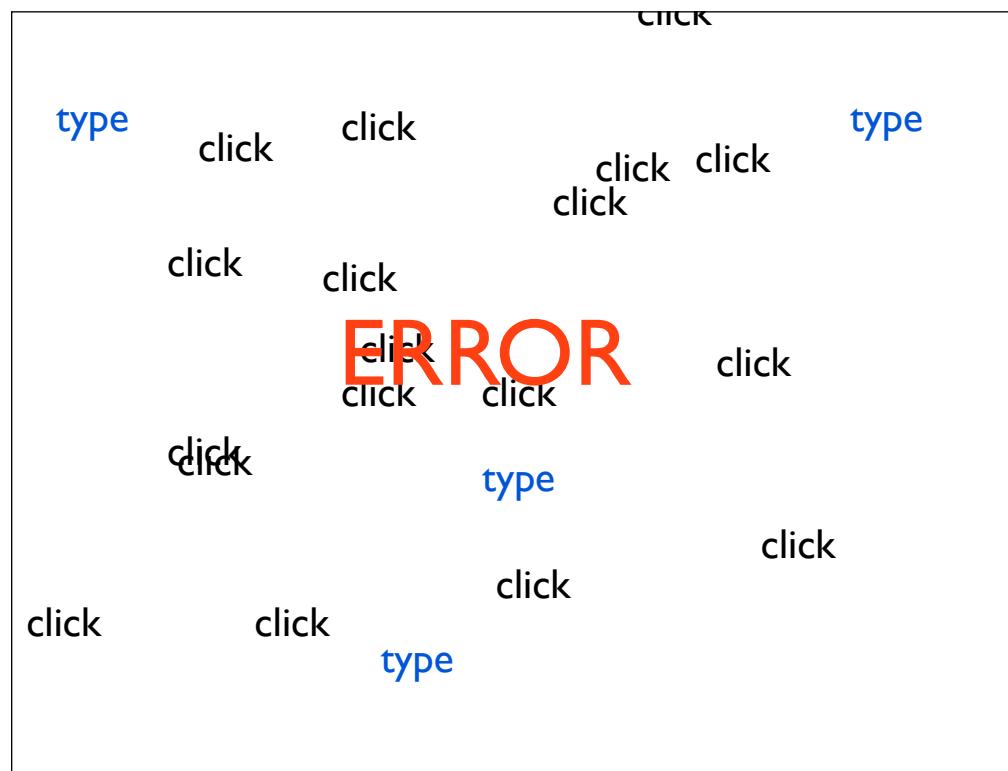
2.Automated acceptance tests

Cucumber or the test runner of your choice

Selenium or Sahi  
Capybara



- 1.Unit and integration tests
- 2.Automated acceptance tests
- 3.User acceptance tests**



- 1.Unit and integration tests
- 2.Automated acceptance tests
- 3.User acceptance tests**

- 1.Unit and integration tests
- 2.Automated acceptance tests
- 2a.Deploy
- 3.User acceptance tests



Deployment

Brian – champion (“I deploy all the time – I GUESS you could automate”)  
Srushti – skeptic

Super easy in Ruby  
(those Java guys are suckers)

```
$ ssh root@production.com  
prod:~ root# cd /var/www/site  
prod:~ root# git pull a31a0eb569ba2  
prod:~ root# bundle install  
prod:~ root# rake db:migrate  
prod:~ root# rake jammit:package  
prod:~ root# touch tmp/restart.txt
```

```
$ cap deploy production
```

**Why automate?**

Digression: The Rails deploy model

```
$ ssh root@production.com  
prod:~ root# git clone git@scm:myapp.git /var/www/site/a31a0eb569ba2  
prod:~ root# ln -s /var/www/site/current /var/www/site/a31a0eb569ba2  
prod:~ root# bundle install  
prod:~ root# rake db:migrate  
prod:~ root# rake jammit:package  
prod:~ root# touch tmp/restart.txt
```

```
$ ssh root@production.com  
prod:~ root# git clone git@scm:myapp.git /var/www/site/a31a0eb569ba2  
prod:~ root# ln -s /var/www/site/current /var/www/site/a31a0eb569ba2  
prod:~ root# bundle install  
prod:~ root# rake db:migrate  
prod:~ root# rake jammit:package  
prod:~ root# touch tmp/restart.txt
```

```
ci-server:~/app/ $ bundle install --deployment  
ci-server:~/app/ $ rake jammit:package  
ci-server:~/app/ $ tar zxvf app-<SHA>.tgz *
```

**RVM**

Not for production

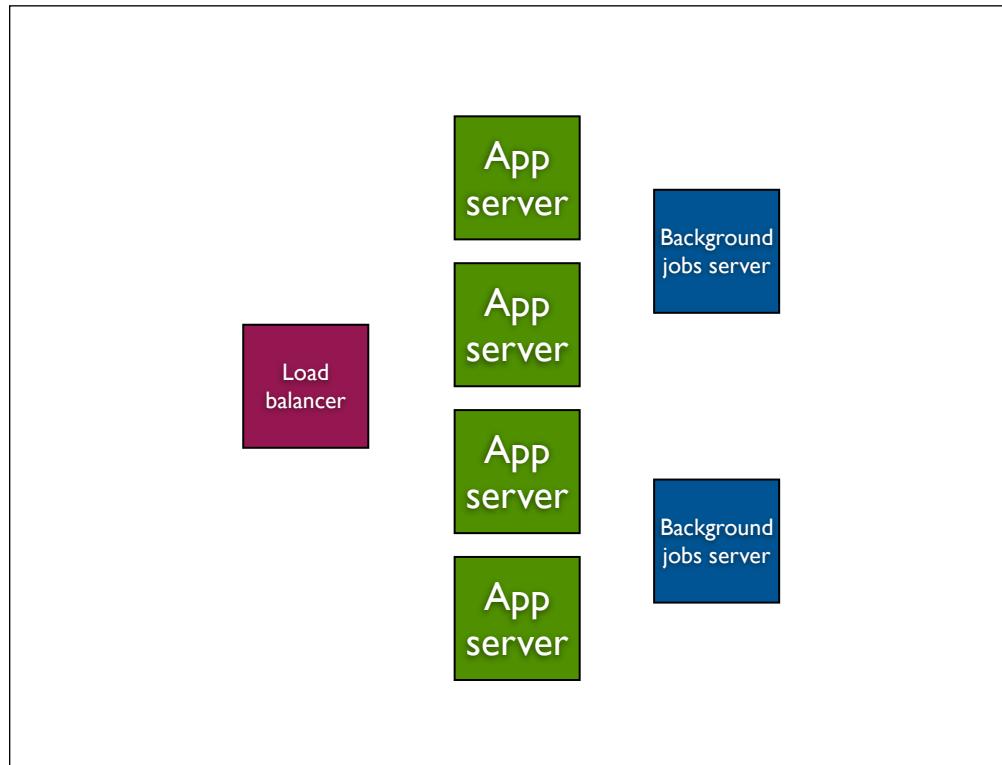
```
$ cat ~/projecta/.rvmrc  
rvm jruby@projecta
```

```
$ cat ~/projecta/.rvmrc  
rvm ree@projectb
```

```
$ cat ~/projecta/.rvmrc  
rvm 1.9.1@projectc
```

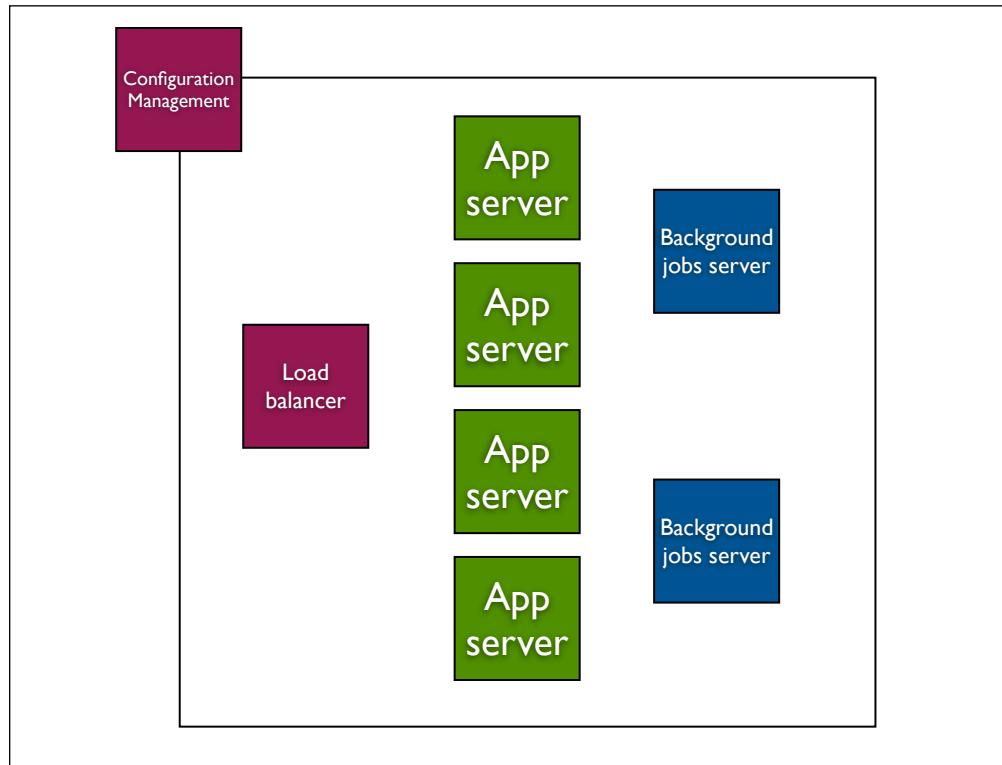
Environments

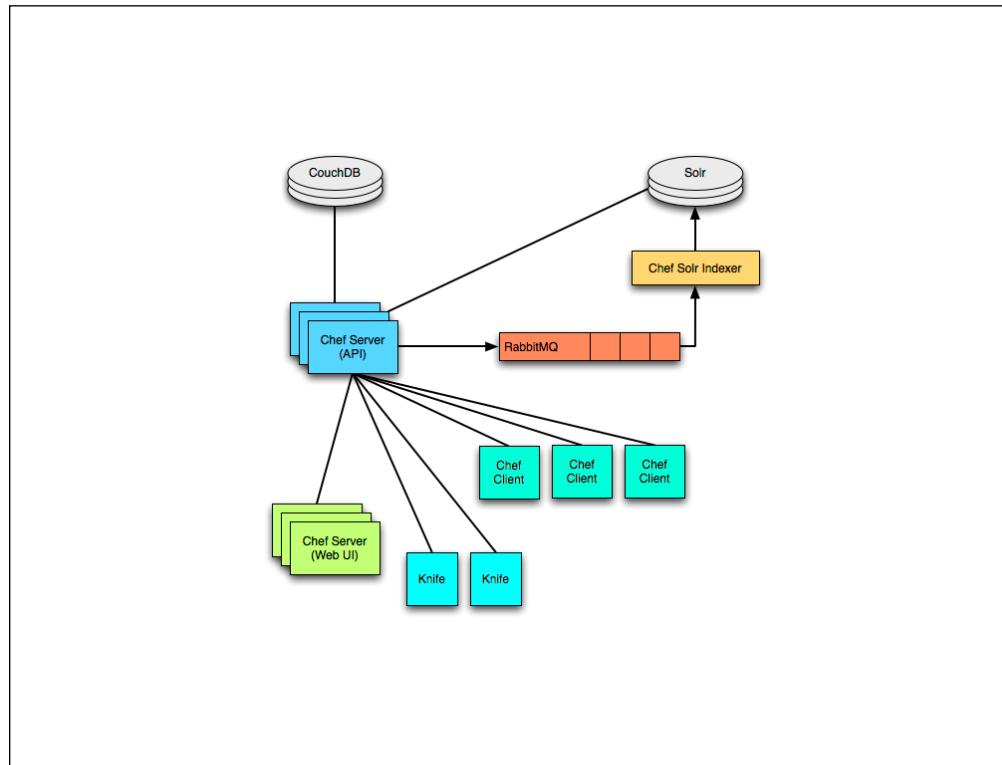
Brian – champion, Srushti – skeptic



~~Chef~~  
Puppet









Peace in the kingdom

Brian – more rapid, then convinced to be slower  
Srushti – slower, then convinced to be more rapid

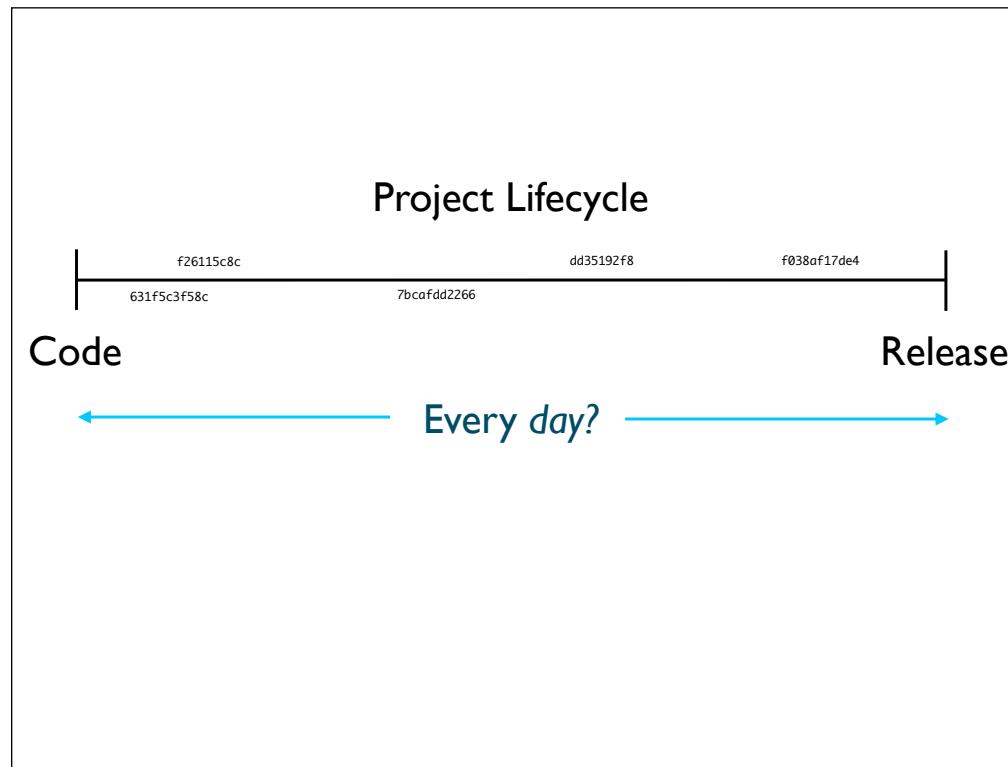
*How* often?

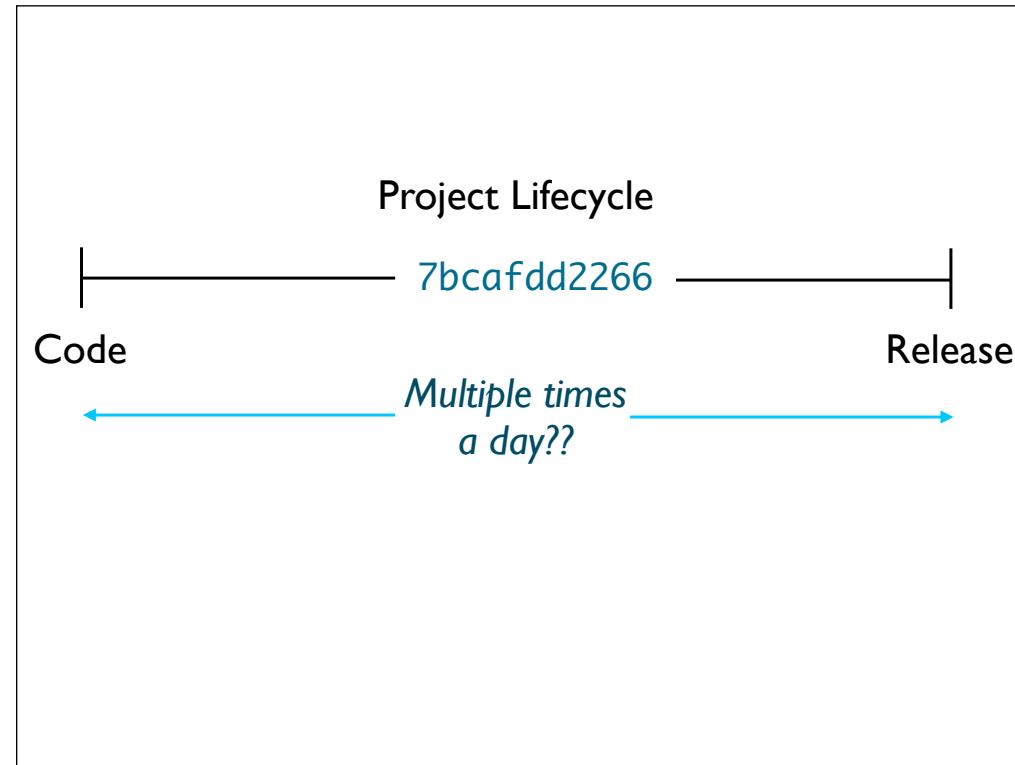
# Project Lifecycle



## Project Lifecycle







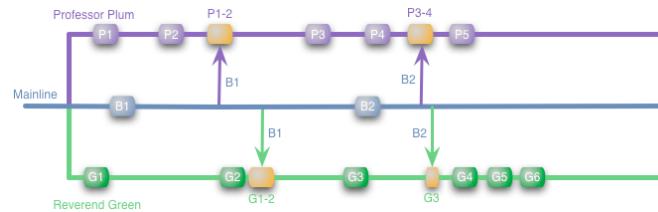
Features long in development need branches

## Branch by abstraction ("if this feature is enabled, show this link")

```
<% if feature?(:masala) %>
  <%= render :partial => "dosas/masala",
             :locals => { :dosa => @dosas.masala.first } %>
<% else %>
  <%= render :partial => "dosas/plain",
             :locals => { :dosa => @dosas.first } %>
<% end %>
```

This doesn't work for everything:  
big features, config changes

## Branch by source control ("fork me")



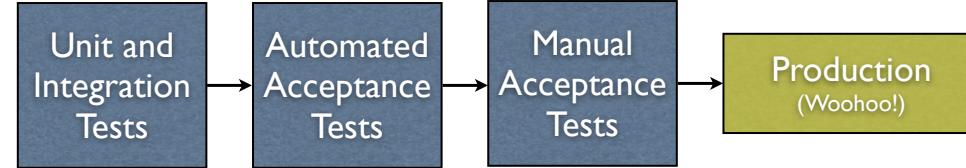
<http://martinfowler.com/bliki/FeatureBranch.html>

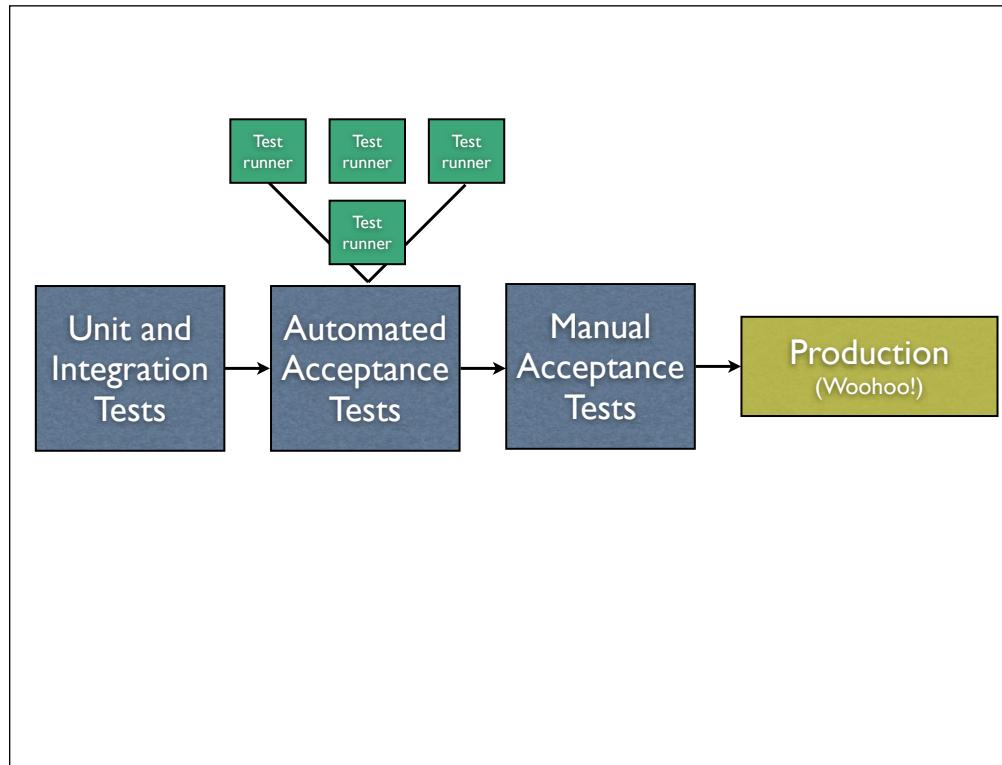
## The Build Pipeline

Brian – skeptic because wants slower deploys, Srushti – champion

“...an automated manifestation of your process for getting software from version control into the hands of your users.”

- Jez Humble and David Farley, *Continuous Delivery*



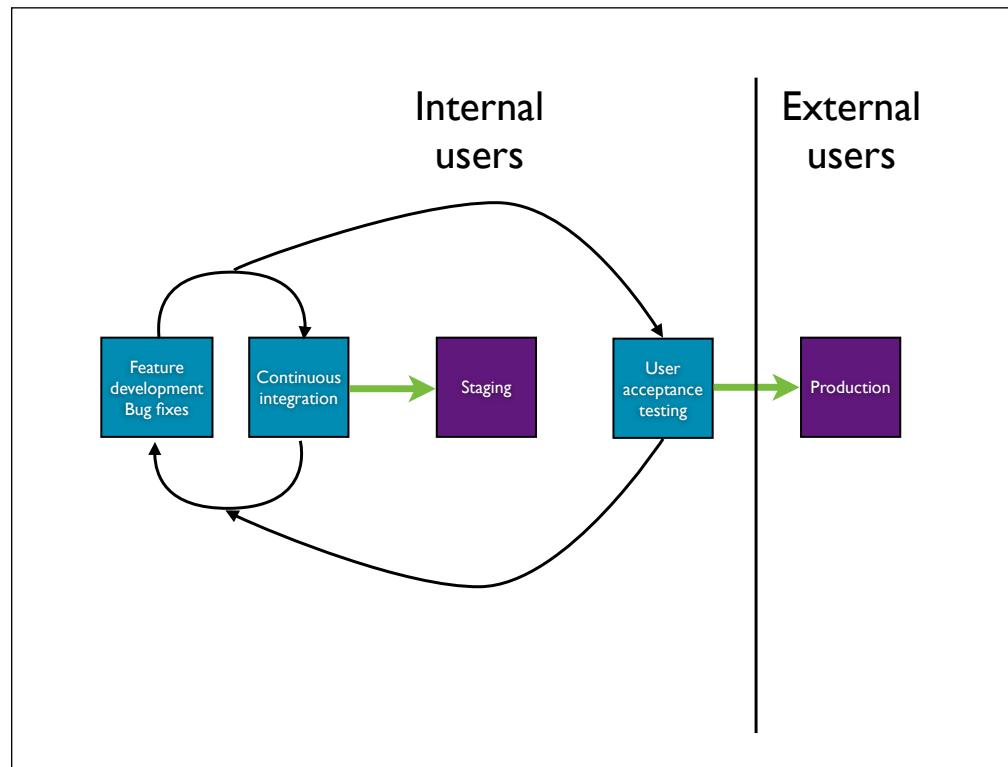


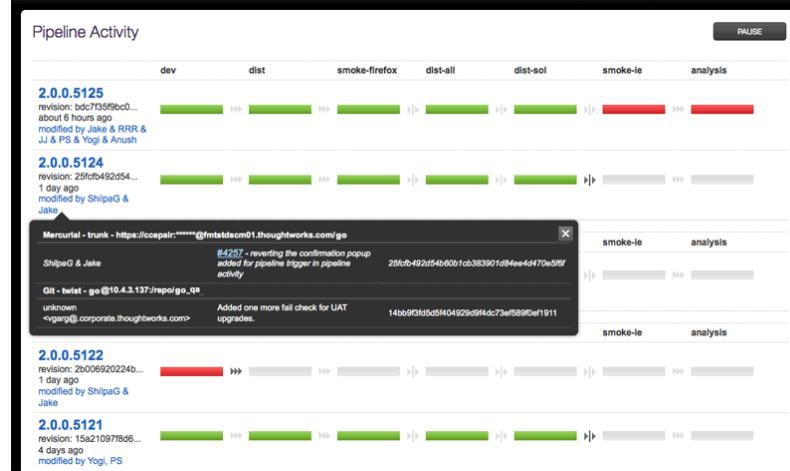
“I see CI as primarily giving birth to a release candidate at each commit. The job of the CI system and deployment process is to disprove the production-readiness of a release candidate. This model relies on the need to have some mainline that represents the current shared, most up to date picture of complete.”

- David Farley

Automate until human intervention = *decision*

DEPLOY





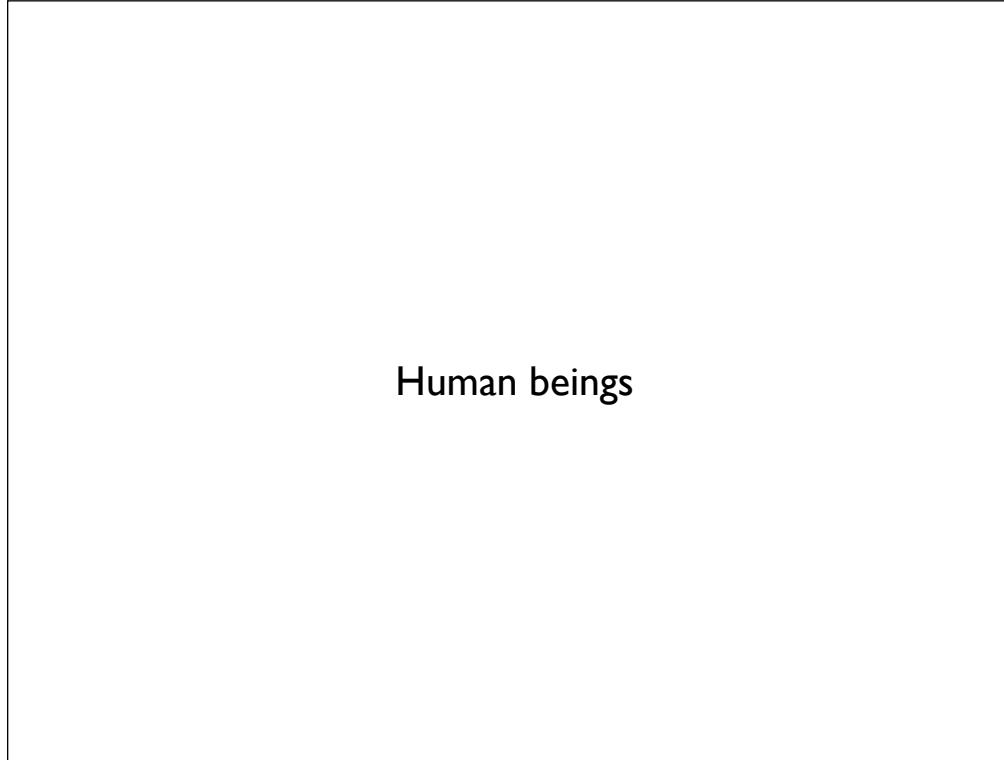
# Goldberg

Lightweight Continuous Integration in Ruby

Project Name	Last Built at	Action
<a href="#">goldberg</a> (1.9.2)	about 1 hour ago	force build
<a href="#">qTestable</a> (1.9.2)	20 days ago	force build
<a href="#">wrest</a> (1.9.2)	about 22 hours ago	force build
<a href="#">rspec-http</a> (1.9.2)	20 days ago	force build
<a href="#">stiritz</a> (1.9.2)	about 6 hours ago	force build
<a href="#">rspec-mocks</a> (ruby)	1 minute ago	force build
<a href="#">wrest19</a> (1.9.2)	about 22 hours ago	force build
<a href="#">pox_paginate</a> (1.9.2)	19 days ago	force build



Etsy – will open source later

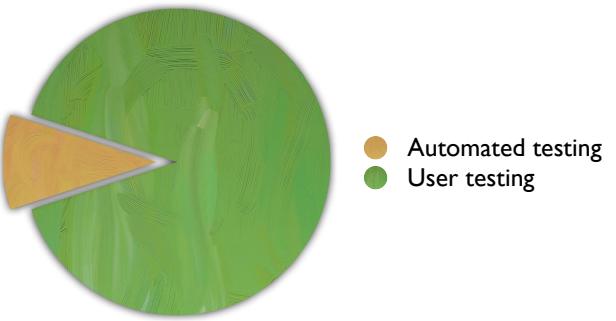


Human beings

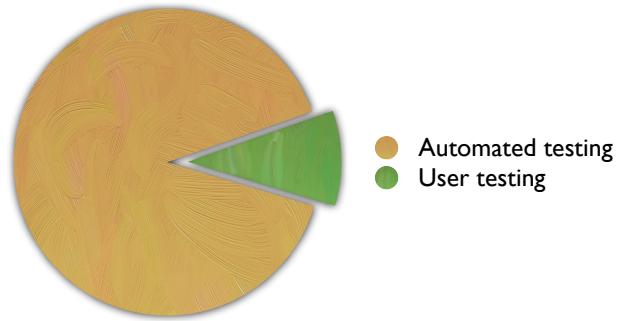
Brian – skeptic (“I used to have an army of testers”)  
Srushti – champion



Real testing is hard  
Let computers do the simple testing  
Smart humans for smart testing



Traditional projects



Agilify!



Agilify!

# Questions?

 Srushti Ambekallu  
42 @srshti  
github.com/srushti

Brian Guthrie  
**Thought**Works® @bguthrie  
github.com/bguthrie