move fast
break
&
nothing
Move fast and break things

– Mark Zuckerberg, Facebook
“Move fast and fix things”

– Mark Zuckerberg, Facebook
Move fast with stable infra

— Mark Boringberg, Facebook
Move fast and be bold

— Miguel Velazquez, Facebook
“Drop the ‘the’ – it’s cleaner.”

– JUSTIN TIMBERLAKE 💙 SWOOOOOOON
moving quickly and safely is good philosophy
what does it mean in practice?
got me thinking about building environments where success is a byproduct
code, process & talk

3 areas
code
move fast
people like fast better product
people like fast see what works
people like fast momentum
and break things
break things

fix later and... stuff

and

break things
some things you can’t break
billing · permissions · upgrades · migrations
how do you move fast in dicey code?
we redid github’s permissions code
hella scary.
.01% errors = unacceptable
0 errors = mandatory
how do we ship this fast + safe?
tests are great,
of course
production can differ from what your tests expect
production can differ from what you even expect
ensure that your code doesn’t change production behavior
parallel code paths
user.old_code always gets run
A request is made, with both `user.old_code` and `user.new_code` always getting run.
user.old_code always gets run
user.new_code also gets run
response

request

user.
run both, compare
science "new-auth" do |e|
  e.control { user.slow_auth }
  e.candidate { user.fast_auth }
end
science "new-auth" do | e |
  e.control { user.slow_auth }
  e.candidate { user.fast_auth }
end
always runs and returns your result
can be run as a percentage

```ruby
science "new-auth" do |e|
  e.control { user.slow_auth }
  e.candidate { user.fast_auth }
end
```
science “new-auth” do | e |
  e.control { user.slow_auth }
  e.candidate { user.fast_auth }
end

collects the two results
75th and 99th percentile performance
quickly iterate until you can prove safety
what do you have today that you can already use?
each layer of process added is expensive
instead grow
process laterally
tests rule everything around me
ci as code maintenance
failing test: removing html class without cleaning css
failing test: removing css without removing html class
failing test: adding `<img>` that's not on our cdn
failing test: invalid css (scss linting)
automates gruntwork out of code review
we test blog posts, too
failing test: images off our cdn
failing test: image non-retina or too big
failing test: using an oft-used phrase
what's important?
feature launches need:

deployers · designers · qa
· product managers · lawyers · marketing · ops
text files are rad
anpp
(apple new product process)
anpp (apple new product process) giant-ass checklist
anpp (apple new product process)

lists at tim-level and small team-level
anpp (Apple New Product Process) happens as very first step
ship lists
ship lists happens often as last step
ship lists all the todos for teams for their +1
simple todo lists make you think less
ownership
don’t throw code over the wall
tie code to writers
notifies you if you screw up
tie code to writers
Hubot
Exceptions have recently elevated on github (18 exceptions) in the last 3 minutes. @holman was the last person to deploy at 02:45 pm PDT (-0700). Care to check it out in haystack?
authors know their own code
authors should fix their bugs
How can your process help out?
tie code to writers
tie code to writers
chrome has owner files
tie code to writers
dictates responsibility
tie code to writers
makes mentors visible

tie code to writers
tie code to writers
class BranchesController
  areas_of_responsibility :git
end

tie code to writers
GitHub / GitHub::TimeoutMiddleware::RequestTimeout

Overview

Dashboard

Firehose

Experienced by you

Latest Deploys

- stargazer-spam deployed by pengwynn 28 minutes ago
- catch-transaction-race-condition-regress... deployed by keleigle an hour ago
- API-direct-org-team-repos deployed by keleigle 2 hours ago
- diff_toc_renamed_without_changes deployed by ymereli 2 hours ago
- master deployed by dbuskink 3 hours ago

Searches

- Rails 3
- 404 Errors

Search...

Create issue on github/github

GitHub::TimeoutMiddleware::RequestTimeout in FilesController#file_list
September 26, 2014 17:31:39

github/github#33514

GitHub::TimeoutMiddleware::RequestTimeout in BertRPC with fast 'Action#read'
September 19, 2014 21:16:19

github/github#33228

GitHub::TimeoutMiddleware::RequestTimeout in Allow cross referencing a PR
September 25, 2014 21:08:26

github/haystack#897

Dashboard

Frequent Users

<table>
<thead>
<tr>
<th>User</th>
<th>Count</th>
<th>Repository</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>hubot</td>
<td>25</td>
<td>Homebrew/homebrew</td>
<td>3217</td>
</tr>
<tr>
<td>boxen</td>
<td>21</td>
<td>CocoaPods/Specs</td>
<td>804</td>
</tr>
<tr>
<td>github</td>
<td>20</td>
<td>torvalds/linux</td>
<td>598</td>
</tr>
</tbody>
</table>

Frequent Repos

<table>
<thead>
<tr>
<th>Repository</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homebrew/homebrew</td>
<td>3217</td>
</tr>
<tr>
<td>CocoaPods/Specs</td>
<td>804</td>
</tr>
<tr>
<td>torvalds/linux</td>
<td>598</td>
</tr>
</tbody>
</table>

Frequent Routes

<table>
<thead>
<tr>
<th>Route</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>github/fs103-cp-1-prd</td>
<td>3421</td>
</tr>
<tr>
<td>github/fs122-cp-1-prd</td>
<td>1455</td>
</tr>
<tr>
<td>github/fs125-cp-1-prd</td>
<td>821</td>
</tr>
</tbody>
</table>

Frequent Servers

<table>
<thead>
<tr>
<th>Server</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>github/fs103-cp-1-prd</td>
<td>240</td>
</tr>
<tr>
<td>github/fs122-cp-1-prd</td>
<td>227</td>
</tr>
<tr>
<td>github/fs125-cp-1-prd</td>
<td>199</td>
</tr>
</tbody>
</table>
Tie the code to writers.

Issues get assigned to the appropriate team.
Hello
is it
BEER you're
looking for?
more communication!
yayyyyyyyyyyyyyycrap
most things aren’t emergencies
most things aren’t emergencies

be mindful of time
be mindful of time

measure impact of your actions
imagine:
1. page a coworker for help
2. they get woken up
3. their phone takes a selfie of them
4. selfie gets posted into chat
would this make you respect your coworker’s time more?
add empathy to your process
empathy stems from being exposed to real pain
everyone should do some support
there’s a difference between reading about it & helping with it
institutional teaching
[we] have a responsibility to be teachers—that this should be a central part of [our] jobs.

- Ed Catmull, Creativity, Inc.
...it’s just logic that someday we won’t be here.

— Ed Catmull, Creativity, Inc.
more productive if workers are smart
how do you teach without making it lame?
chatops: learning by osmosis
app crashes.

Here's a normal flow:
app crashes. Here's a normal flow:

1. employee gets paged
2. they ssh into... something
3. they fix it... somehow
no visibility means no teaching
no visibility means no improvement
app crashes. Here's a better flow:

1. employee gets paged
2. they manage it in a chat room
3. they fix it and people can watch
visibility means teaching
visibility means improvement
real-life debugging
github’s wifi is down
floor two, san francisco
(right now!)
Wifi on floor two is affected. Status of wired networks is currently unknown.
I appeared to connect to an AP just fine in the 2nd floor cafe, but didn't get any DHCP responses. tshark log of before/after enabling wifi below:

```
arp query 0x0000  PTR _airport._tcp.local, "QM" question PTR _airplay._tcp.local, "QM" quest
ard query 0x0000  PTR _airport._tcp.local, "QM" question PTR _airplay._tcp.local, "QM" quest
ard query 0x0000  PTR _airport._tcp.local, "QM" question PTR _airplay._tcp.local, "QM" quest
ard query 0x0000  PTR _airport._tcp.local, "QM" question PTR _airplay._tcp.local, "QM" quest
ard query 0x0000  PTR _airport._tcp.local, "QM" question PTR _airplay._tcp.local, "QM" quest
ard query 0x0000  PTR _airport._tcp.local, "QM" question PTR _airplay._tcp.local, "QM" quest
CP Request  - Transaction ID 0x6834fef0
MpV6 48 Router Solicitation
arp query response 0x0000  SRV, cache flush 0 0 5900 foobar.local TXT, cache flush PTR fooba
ard query response 0x0000  SRV, cache flush 0 0 5900 foobar.local TXT, cache flush PTR fooba
ard query response 0x0000  SRV, cache flush 0 0 5900 foobar.local TXT, cache flush PTR fooba
ard query response 0x0000  SRV, cache flush 0 0 5900 foobar.local TXT, cache flush PTR fooba
ard query response 0x0000  SRV, cache flush 0 0 5900 foobar.local TXT, cache flush PTR fooba
ard query response 0x0000  SRV, cache flush 0 0 5900 foobar.local TXT, cache flush PTR fooba
ard query response 0x0000  PTR foobar.local SRV, cache flush 0 0 22 foobar.local TXT, cache
ard query response 0x0000  PTR foobar.local SRV, cache flush 0 0 22 foobar.local TXT, cache
ard query response 0x0000  PTR foobar.local SRV, cache flush 0 0 22 foobar.local TXT, cache
ard query response 0x0000  PTR foobar.local SRV, cache flush 0 0 22 foobar.local TXT, cache
```
Current floor 2 status:
Now into the wiring closet to see where that leads...

I think I have it isolated to these two switches:

![Image of network switches with LED indicators showing a specific issue]
debug in the open
feedback
the Blue Angels on feedback
two-way street
two-way street

how well do you receive feedback?
How well do you give feedback?

two-way street
improving feedback across an org
moving fast with a degree of caution
need to be fast
need to be safe
takes effort to achieve both