Startup Architecture

how to lean on others to get stuff done

Robbie Clutton

Software Engineer, Pivotal Labs

PIVOTAL LABS

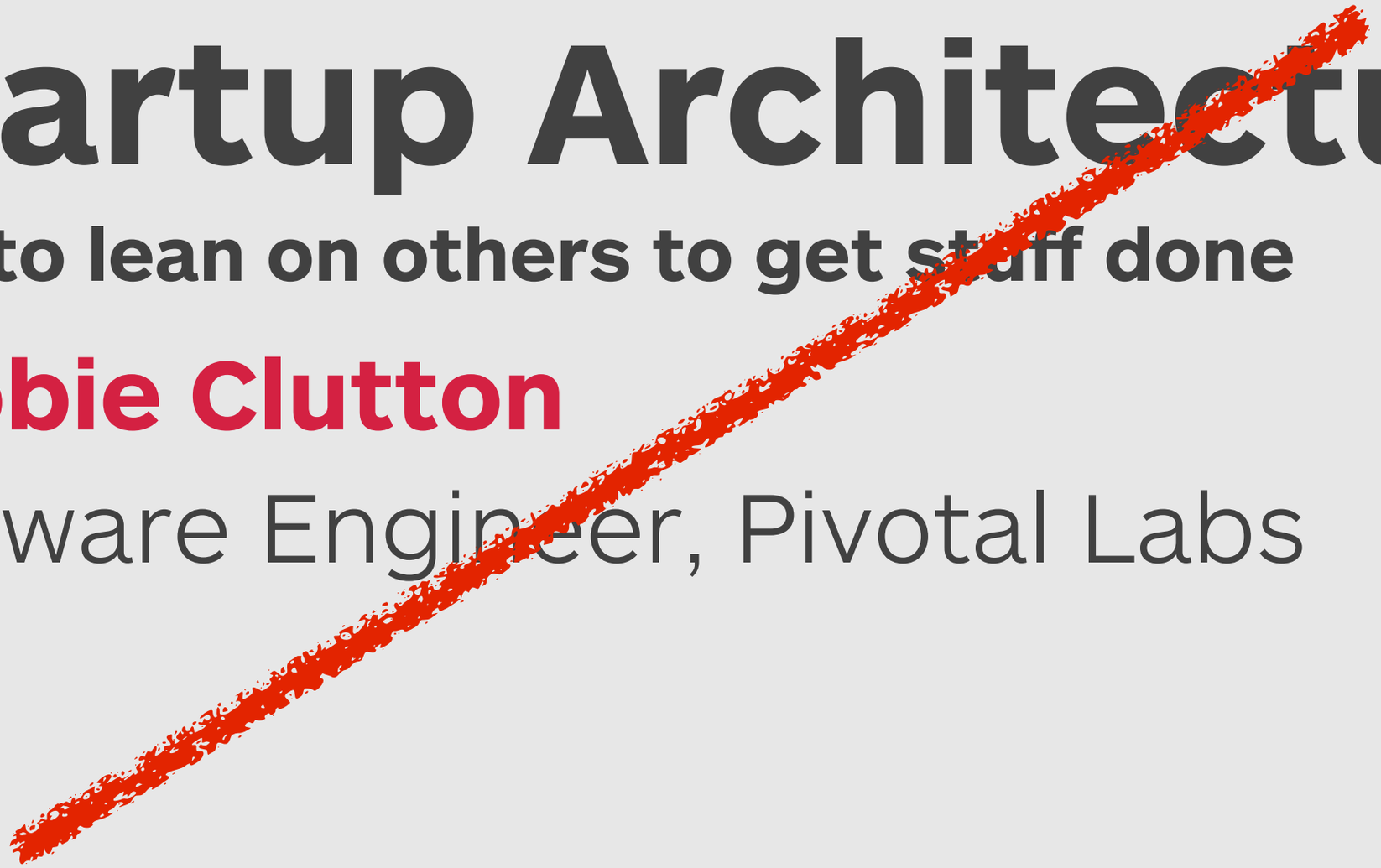
@robb1e

Startup Architecture

how to lean on others to get stuff done

Robbie Clutton

Software Engineer, Pivotal Labs



PIVOTAL LABS

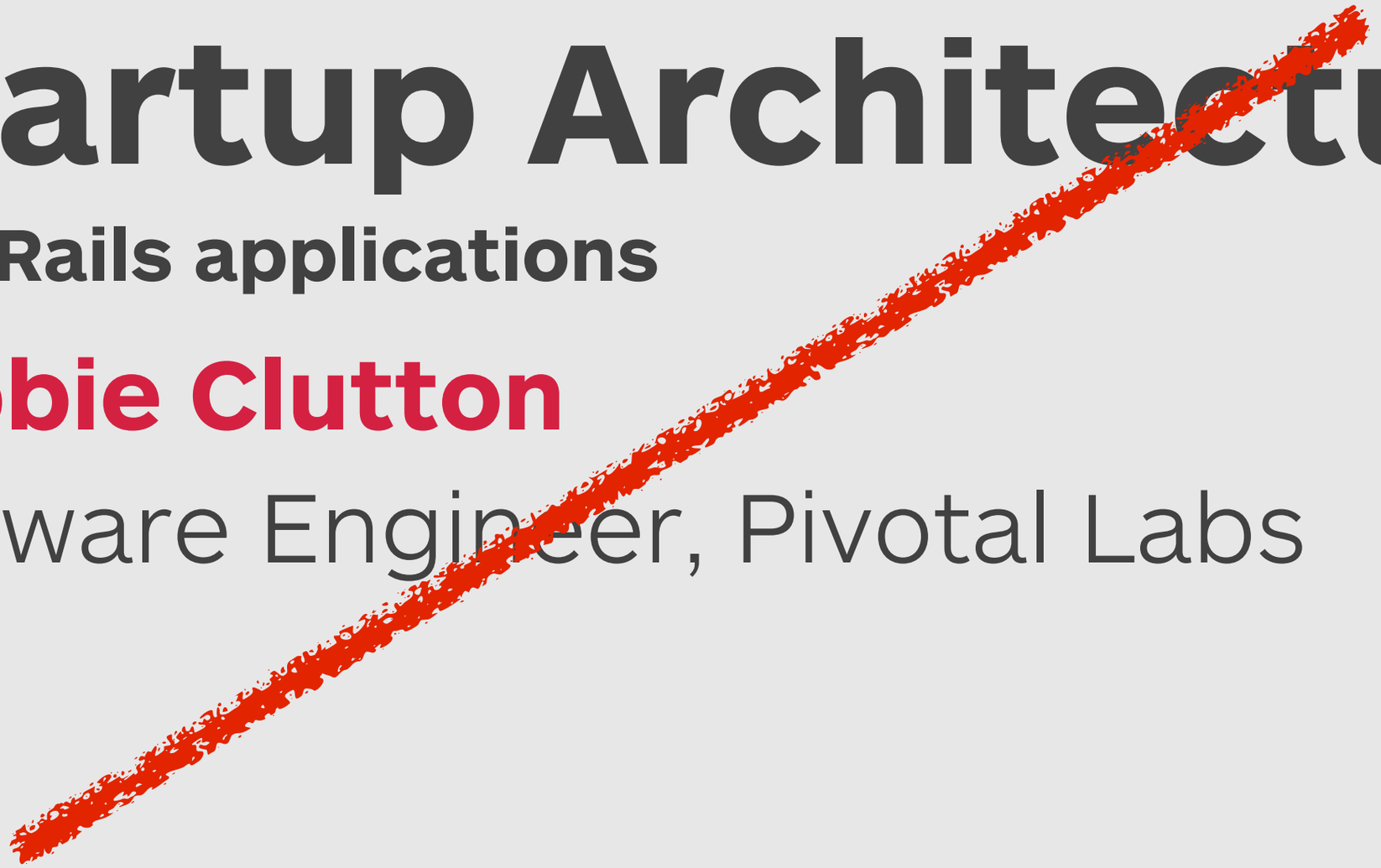
@robb1e

Startup Architecture

with Rails applications

Robbie Clutton

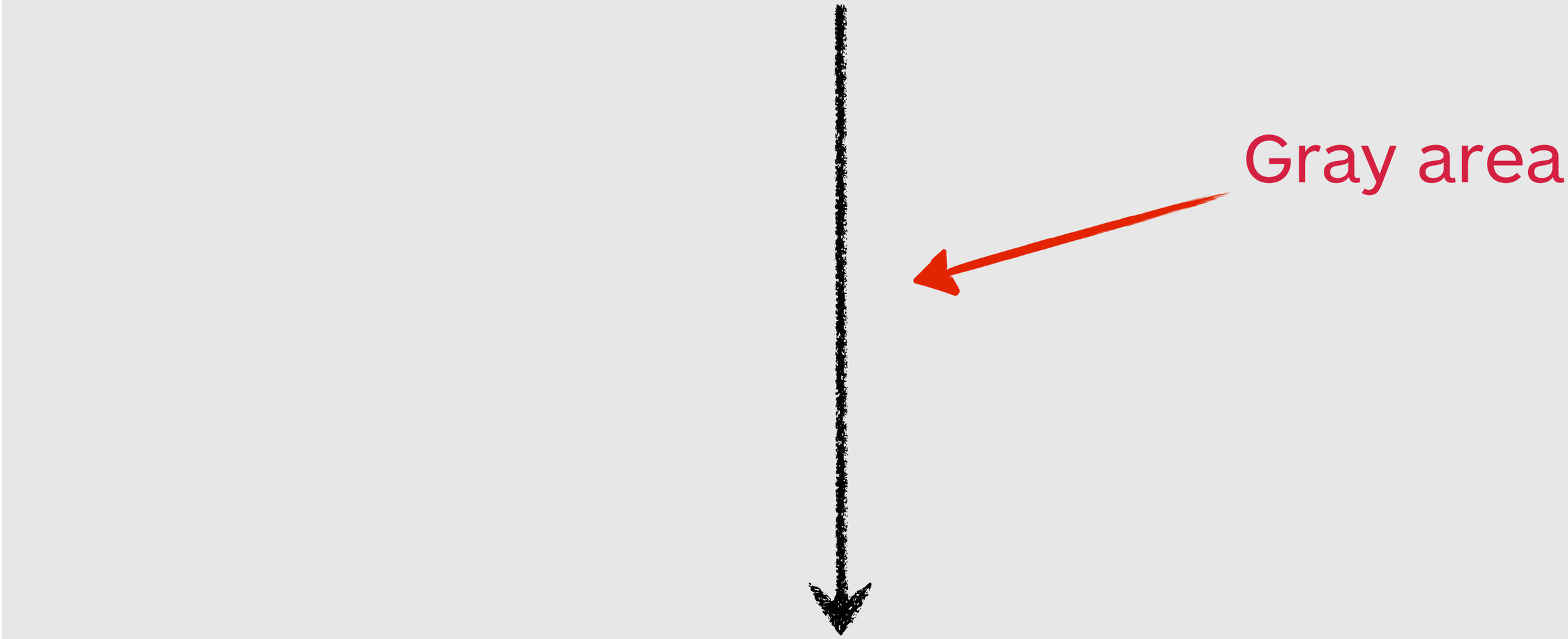
Software Engineer, Pivotal Labs



PIVOTAL LABS

@robb1e

Simple, small web application



Service oriented RESTful/pubsub/LMAX/P2P distributed architecture

Our codebase is 5 years old and too hard to change

We've allowed our design to evolve into a big ball of mud

We'll probably create services at some point, might as well start there

I'm going to design everything up front based on unvaldiated assumptions

Make it work, make it right, make it fast

Kent Beck

Goals

- Features that have hypotheses
- Hypotheses that can be easily validated
- Code that is always production ready
- Code that is easy to change

Startup Architecture

Creating sustainable small architectures

Robbie Clutton

Software Engineer, Pivotal Labs

PIVOTAL LABS

@robb1e

Real stories from colleagues and myself

Names have been changed to protect the innocent

Some stories are pre-production, others are in production

Story

Crazy Egg

10am deploy CrazyEgg

5pm review CrazyEgg

Users clicking headers that are not links

You could feel the users frustration

Lesson

Simple user testing can pay dividends

Tools

- CrazyEgg.com
- UserTesting.com
- SliverbackApp.com
- LeanLaunchLab.com
- Trello.com

Story

Funnels, user testing, hypotheses and validations

Product with wizard like pages which pre-selected default services

Changes to the basket updates price in real-time

Funnel showed massive drop off at a certain step

In person user testing to discover why the drop off was occurring

Create hypothesis to stop users leaving at this junction

Implement change: allow users to use default or create own

Review funnel after deployment

Lesson

Learn what is blocking the users

Tools

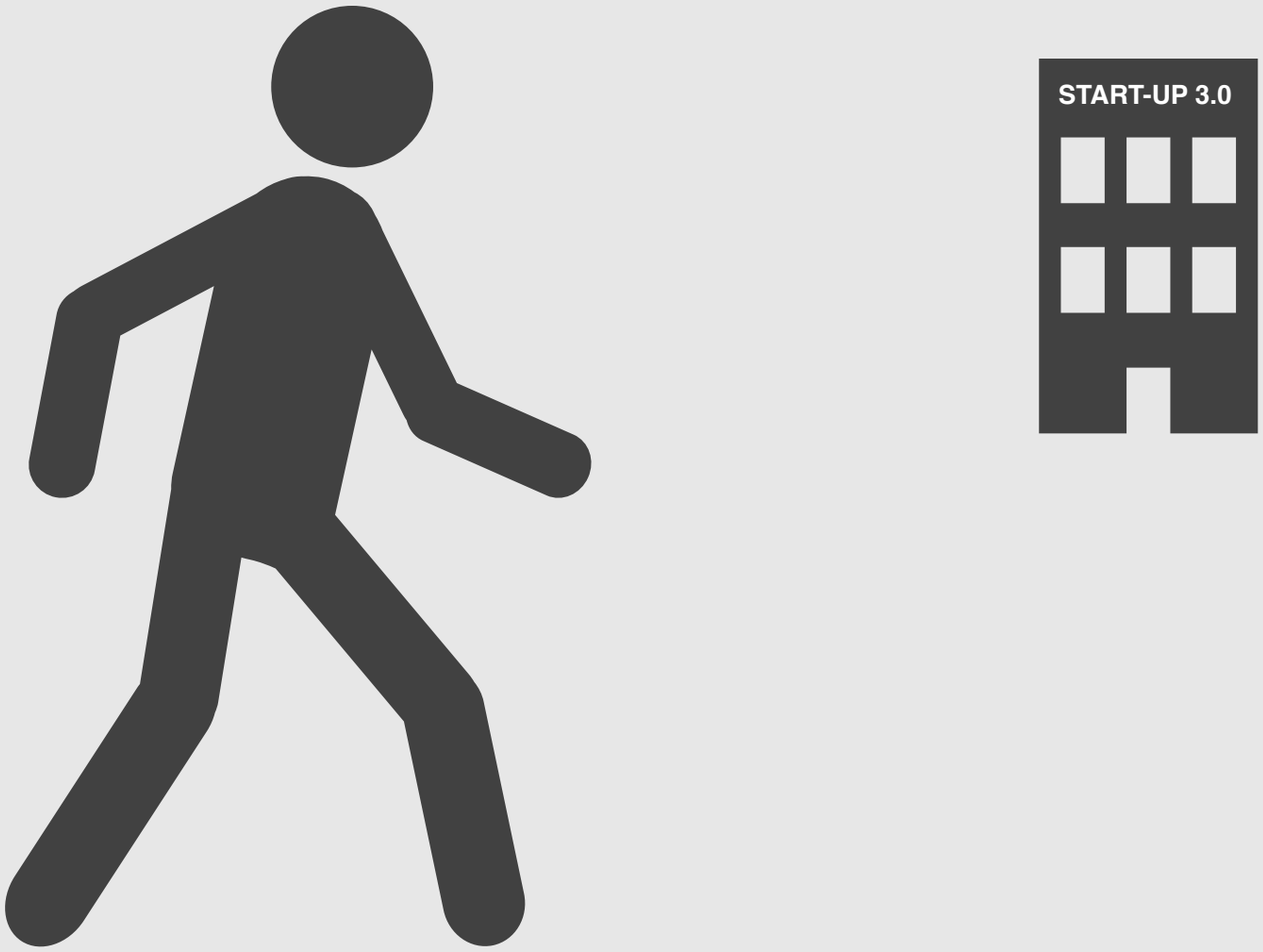
- KissMetrics.com
- StatsD (Etsy)
- Cube (Square)

Take away

Always be validating

Story

You’re gonna need a bigger boat



Dave walks into a new job

Product manager tells you

We need more RAM for the database

Product manager tells you

This report takes 20 minutes to run.

Hmm, ok

There are no indexes


NO
INDEXES

No primary or foreign keys

Needed more RAM so the whole database could fit in memory

Dave cleans up a bit, report now takes 10 seconds to run

Lesson

Use tools to discover simple mistakes

Bonus

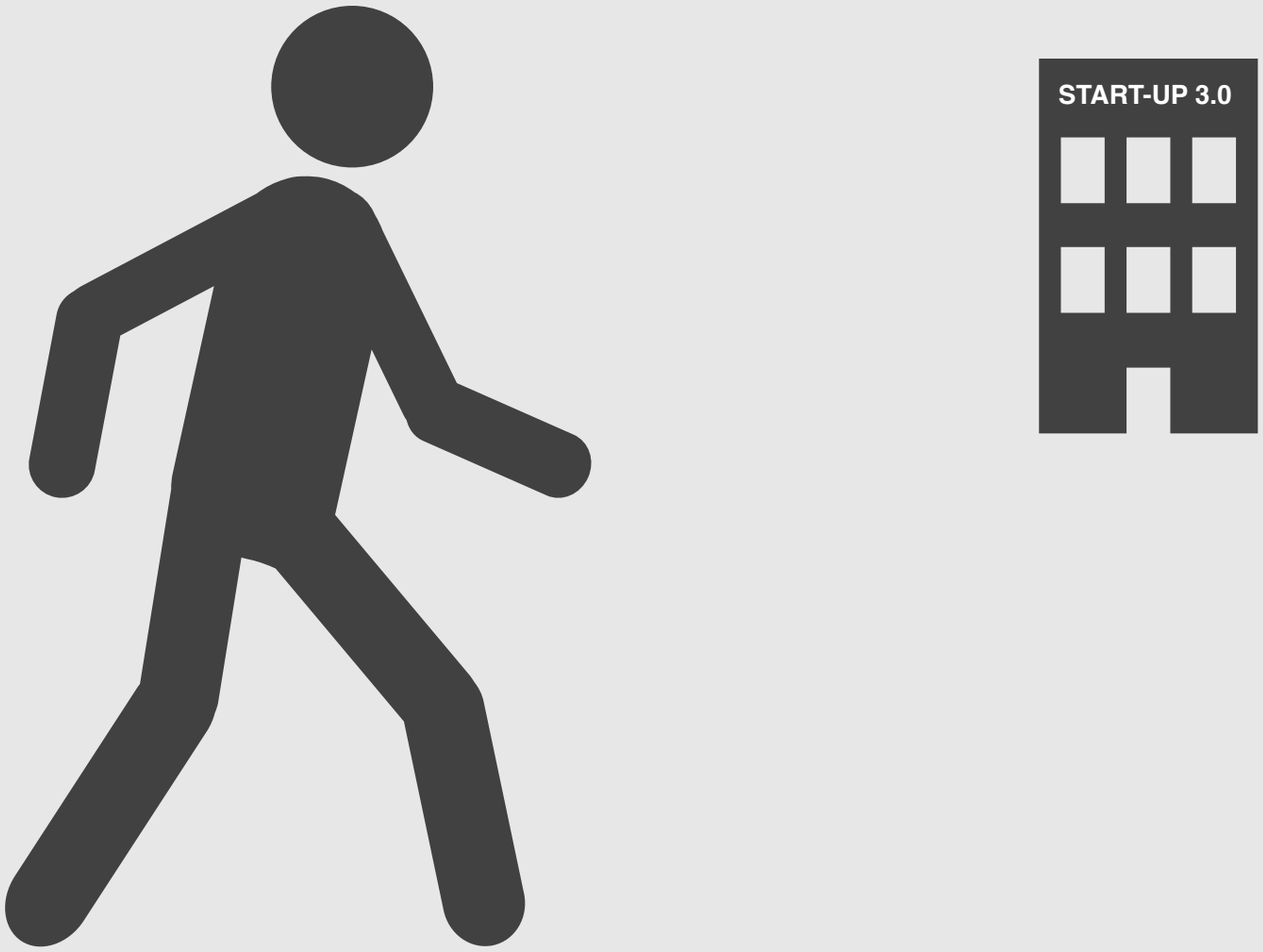
Passing tests don't
imply production quality

Tools

- Rails Best Practices
- SQL Explain
- NewRelic.com

Story

Instrument, refactor, repeat



Dave walks into a new job

Client moving from ColdFusion to Ruby

Yes, there are people still using ColdFusion

Product manager tells you

Ruby is slow and we're going to production next week.

Product manager tells you

We've made a terrible mistake...

Hold on a minute, let's take a look

You say...

Instrument to find slow requests/queries

Refactor slowest query until more performant with green tests

Rinse and repeat until performance has improved enough

Every scaling story:

1. Find the biggest problem
2. Fix the biggest problem
3. Repeat

Paul Hammond, 2012

Bonus

‘Friday afternoon’ performance refactoring can build upon itself

Tools

- NewRelic.com
- CodeClimate.com
- Emma, FindBugs

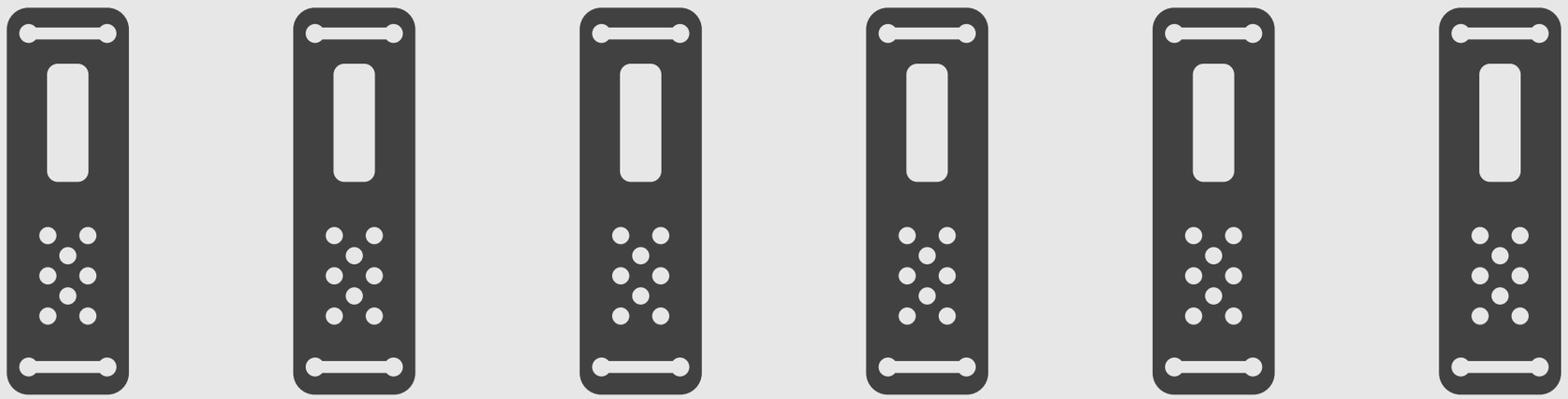
Take away

Use tools to discover improvements

Story

Distributed cache

Website was growing and gaining visitors



Scaling strategy was to add app servers

Each server had the web app and a local cache

Spinning up a new server meant more pressure on the database

Using a distributed cache bought the team time to make improvements

Lesson

Caching can buy significant performance improvements

Tools

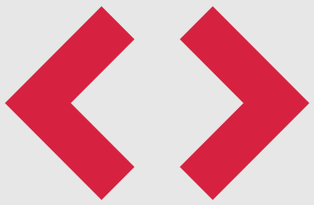
- MemcacheD.org
- Varnish-Cache.org
- Squid-Cache.org

Story

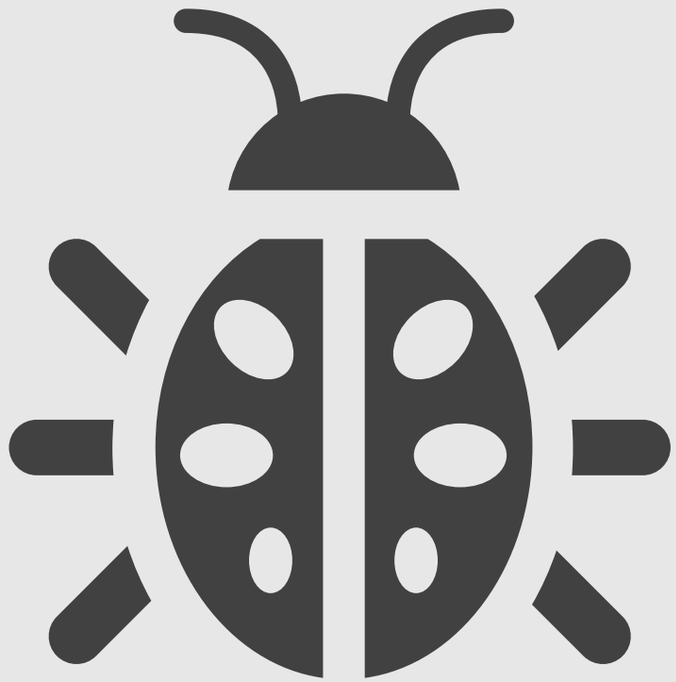
To cache, or not to cache?

Sometimes code speaks to you

This part is slow, let's cache it. Problem solved



But I'm going to invalidate that elsewhere



Collection of widgets being rendered with new and old design

Can’t replicate on staging or locally

Clear ALL the cache

PIVOTAL LABS

@robb1e

Changing the template had not invalidated the entry

"There are only two hard things in Computer Science: cache invalidation and naming things."

- Phil Karlton

Bonus

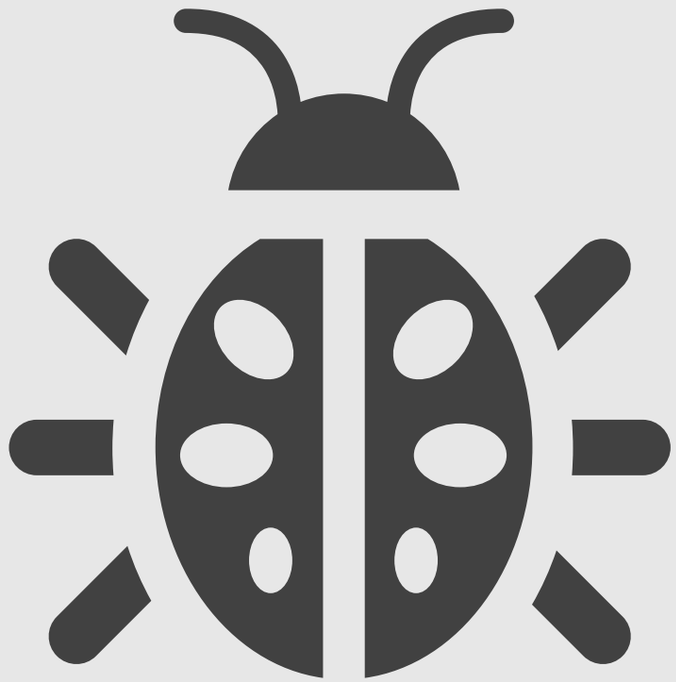
Caching can obsure
poorly written code

Take away

Be careful what you cache

Story

Non-essential work during a request



User registration stopped working

Mailing list provider was down

Exception bubbled up and prevented registering new user

Put mailing list subscription in background job

Lesson

Shorten the request/response cycle

Bonus

When dealing with integrations, some healthy paranoia is a good thing

Tools

- Background workers
- Message Queues
- Threads

Story

A tale of two websites

www.guardian.co.uk

125 requests HTML: 3.7s

1.2MB Loaded: 8.4s

m.guardian.co.uk

44 requests **HTML: 1.68s**

340KB **Loaded: 3.32s**

That's not the result of better SQL or server optimizations

Result of highly tuned client-side Javascript and CSS

PIVOTAL LABS

@robb1e

No (large) Javascript libraries

Not even jQuery

Conditional loading of secondary content

“Optimize front-end performance first, that's where 80% or more of the end-user response time is spent”

- Steve Saunders, 2007

Tools

- Firebug
- Chrome Developer Tools
- Compass
- YSlow
- YUI Compressor

Take away

Perceived performance is more important than actual performance

Story

Was that really the
best use of your time?

During technical due diligence for an acquisition

The company had built their own message queue

No persistence

Didn’t use standard protocol like AMPQ

Not explicitly sending a terminating character would eventually result in the queue crashing

Almost all transactions passed through this queue

Not buying a message queue company

"If it's a core business function - do it yourself, no matter what."

- Joel Spolsky, 2001

Bonus

Time is the most expensive out going

Story

Real-time vs near-time

Trading system which updates users' screen every 10 seconds

Lots of number crunching and message queues

Did some in the field research

Traders only checked values every few minutes

http://www.boldjack.com/wp-content/uploads/2012/01/wall_street4.jpg

This was not high volatile trading

PIVOTAL LABS

@robb1e

Removed message queues and moved to publishing updates to web server directly

Reduced complexity of the product

Always implement things when you actually need them, never when you just foresee that you need them

Ron Jefferies, ~2005

Bonus

‘Real-time’ can mean different things depending on who you talk too

Story

Buy vs build

$50 a month is really expensive for this hosted service

We can build it ourselves and get exactly the features we need

Can you build the widget service yourself in that time?

Are you in the widget business?

The biggest expense for a startup is your time. Not your laptop, not your hosting bill, not your office, but the hours in your day.

Francis Hwang, 2012

Bonus

Focus on your differentiators

Take away

Over engineering is a form of waste

Story

Horizontal Scalability

Guardian Content API is read only and eventually consistent

Used by m., iPhone app, parts of www. and more

Just a simple API over an indexed data store

Each server has it's own data store

Each data store is a replica of an internal master

Lesson

Simple, elegant design can prevent complex architecture creep

Tools

- Solr
- Elastic Search
- MongoDB

Story

Emergency mode

Use of feature switches at Guardian enable ‘super happy fun mode’

Turn features off when site under increased load

Content is king and must be readable at all times

Page pressing enables zero downtime and last fallback

Lesson

Feature flags can offer resilience as well as a way to roll out new features

Take away

Complex should be lots of simple

Allow architecture to evolve

Spend your time wisely

Refactor continuously

“The wrong abstraction is far more damaging than no abstraction at all. Waiting trumps guessing every time”

Sandi Metz

Q/A