

lean ux & lean engineering

applying lean startup principles at enterprise scale at PayPal



@billwscott. sr. dir. user interface engineering @paypal



SF Lean Startup Circle. April 24, 2013.

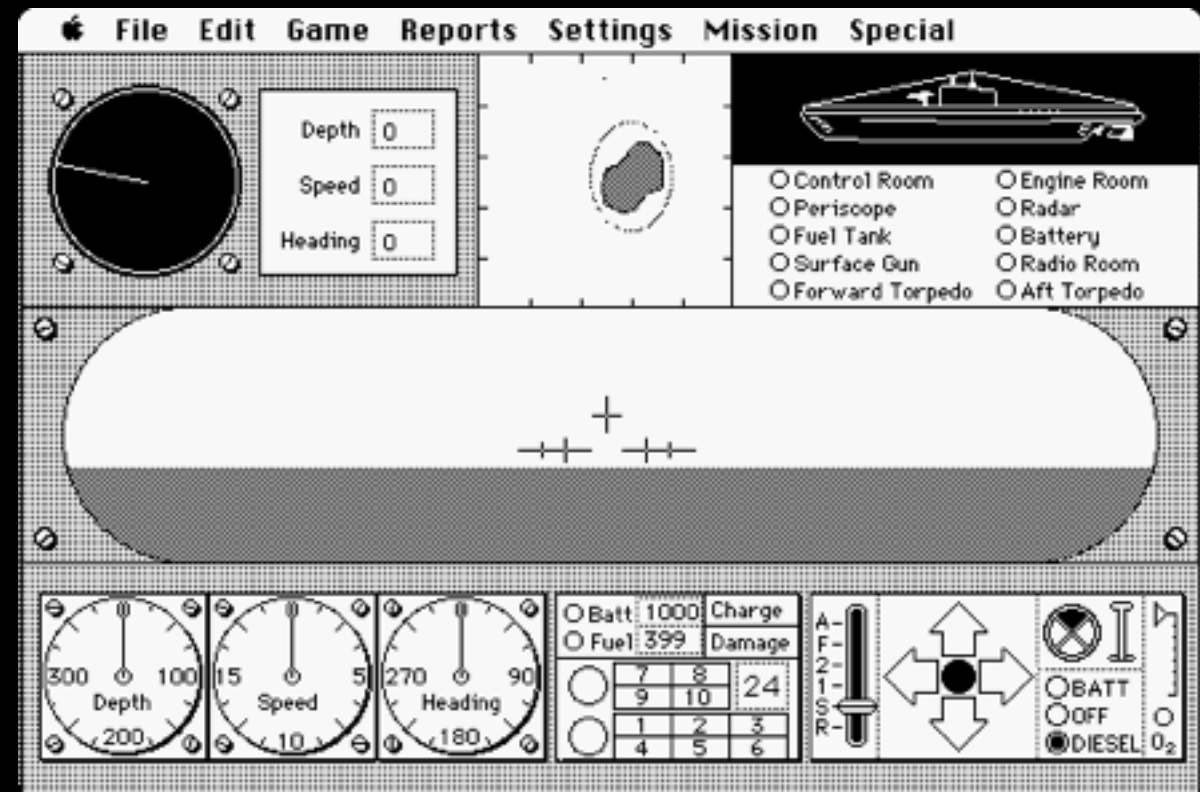
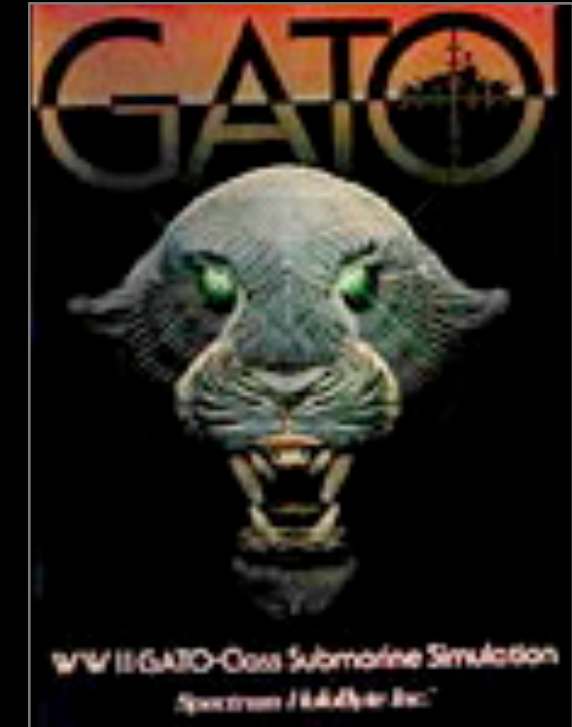
A black and white photograph of a vintage Buick car, likely from the 1930s, featuring a prominent grille with the Buick logo and 'Buick' script. The car has large, round headlights and a spare tire mounted on the side. A crowd of people, including men in suits and hats, is gathered around the car, suggesting a car show or a public event. The image has a torn-edge effect on the right side.

where we have been
how we used to engineer experiences

building experiences
circa 1985

merry band of three. dropped out of
college for semester. it was nirvana.

however...





roll your own everything

(close your eyes & imagine)

no internet. no google. no blogs. no email. no
blogs. no stackoverflow. no github. no twitter.

much of the software era has been about
building from scratch.

of course open source was gaining momentum.
unix. gnu. linux. perl. mozilla.

disconnected delivery experience

deliver to disk then to user

everything was focused on getting it
perfect for stamping on the disk

no user in the loop. experience
happened somewhere down the
supply chain





long shelf life for software

not dynamically updatable

designed for longevity

herculean effort to deliver

engineers ran the asylum

delivery dates drive the experience

BDUF & waterfall

where we should be now

looking at the netflix model

continuous customer feedback (GOOB)

customer metrics drive everything

think it. build it. ship it. tweak it

fail fast. learn fast.

lots of experimentation... build/measure/learn

A photograph of a multi-story building with a curved facade and large windows. A skeleton is hanging from a balcony on the upper floor. In the foreground, a red banner with the word "NETFLIX" in white, bold, sans-serif capital letters is visible.

NETFLIX

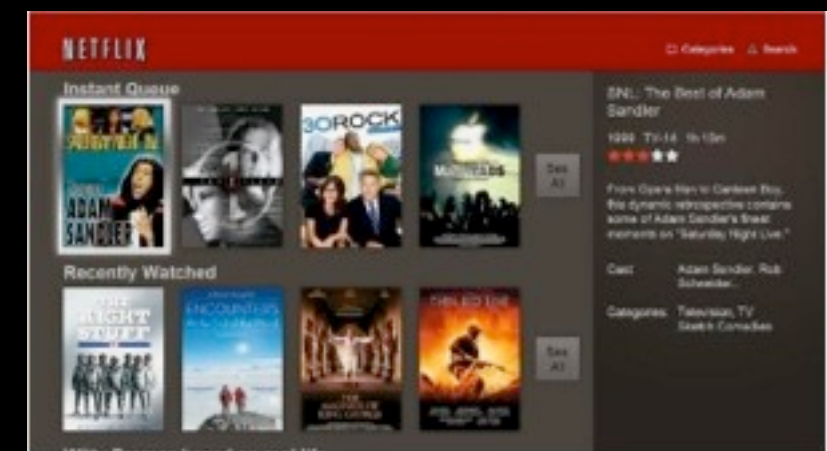
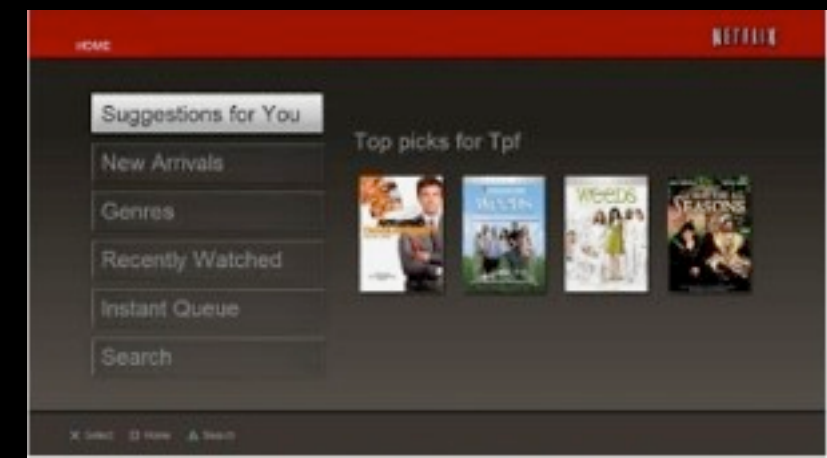
launching the ps3 experience

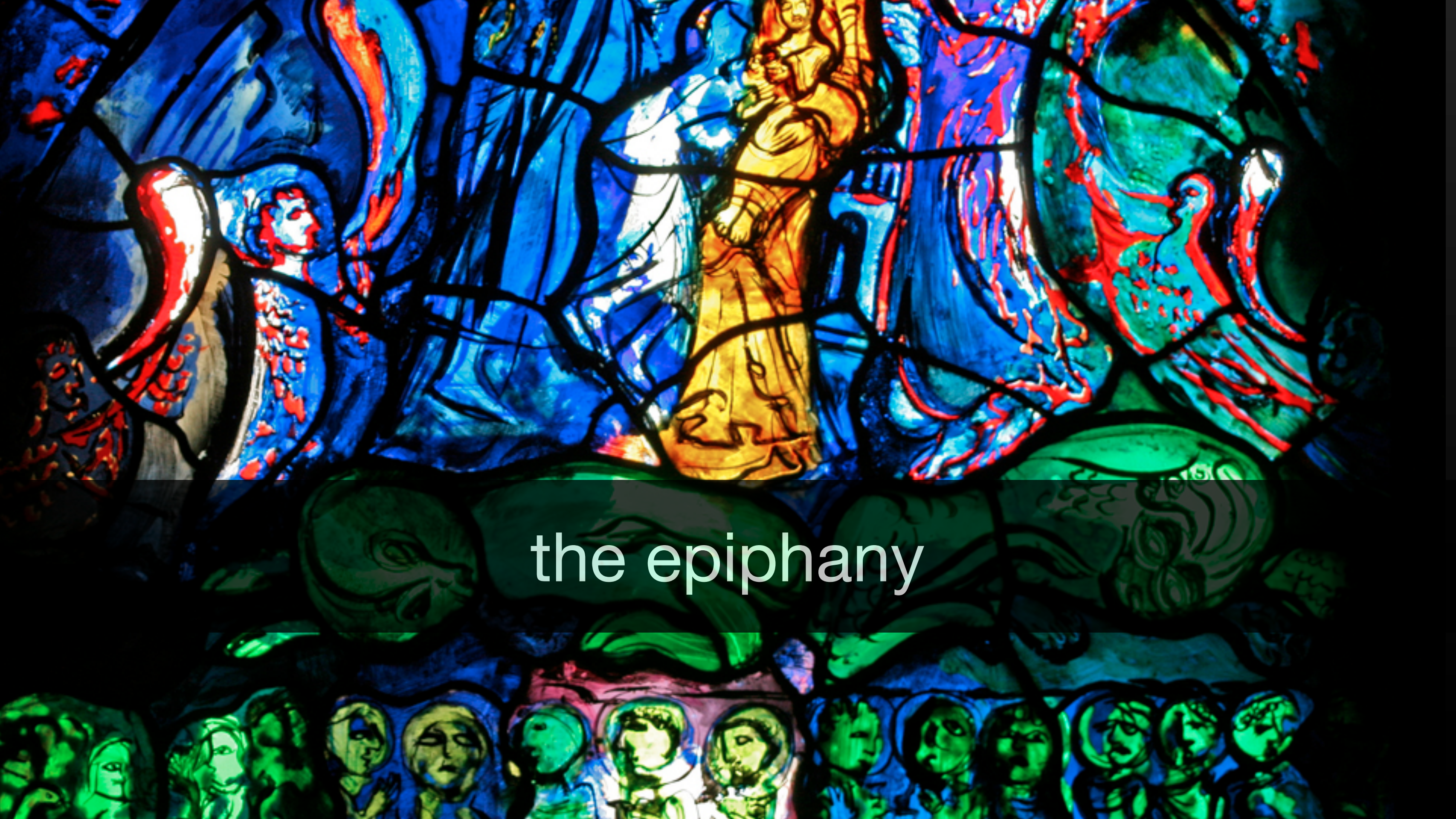
16 different test cells

2 different tech blogs were simultaneously reviewing different experiences

focus was on build/measure/learn

*four
different PS3
experiences
launched
on same day*





the epiphany

you have to engineering for volatility

change is the norm

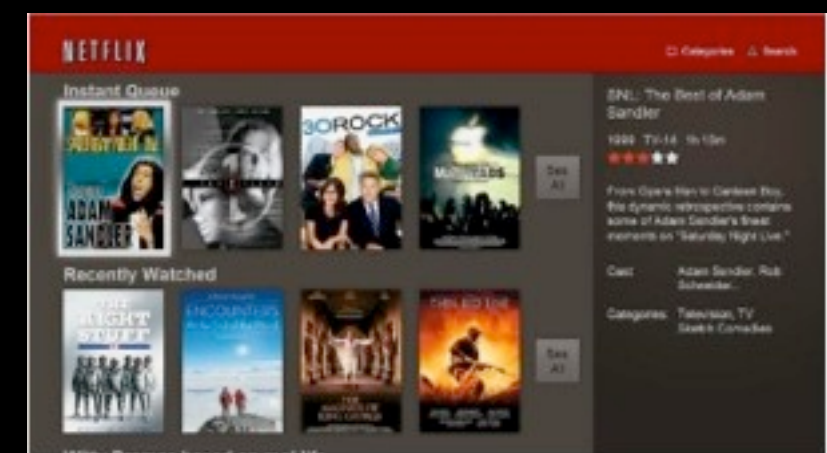
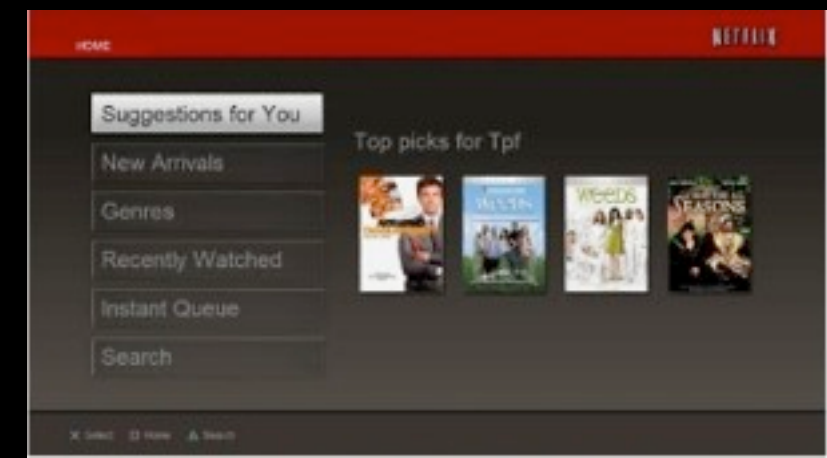
the ui layer is the experimentation layer

experimentation is not a one time event

launching a product is giving birth to the
product. the product's life just begins.

design for throwaway-ability

*majority of the
experience code
written is thrown
away in a year*





paypal vs netflix

contrast this with a large enterprise like paypal (circa 2011)

guess what i found (in 2011)

roll your own. disconnected delivery
experience. culture of long shelf life.
inward focus. risk averse.

new dna inserted

jan 2012

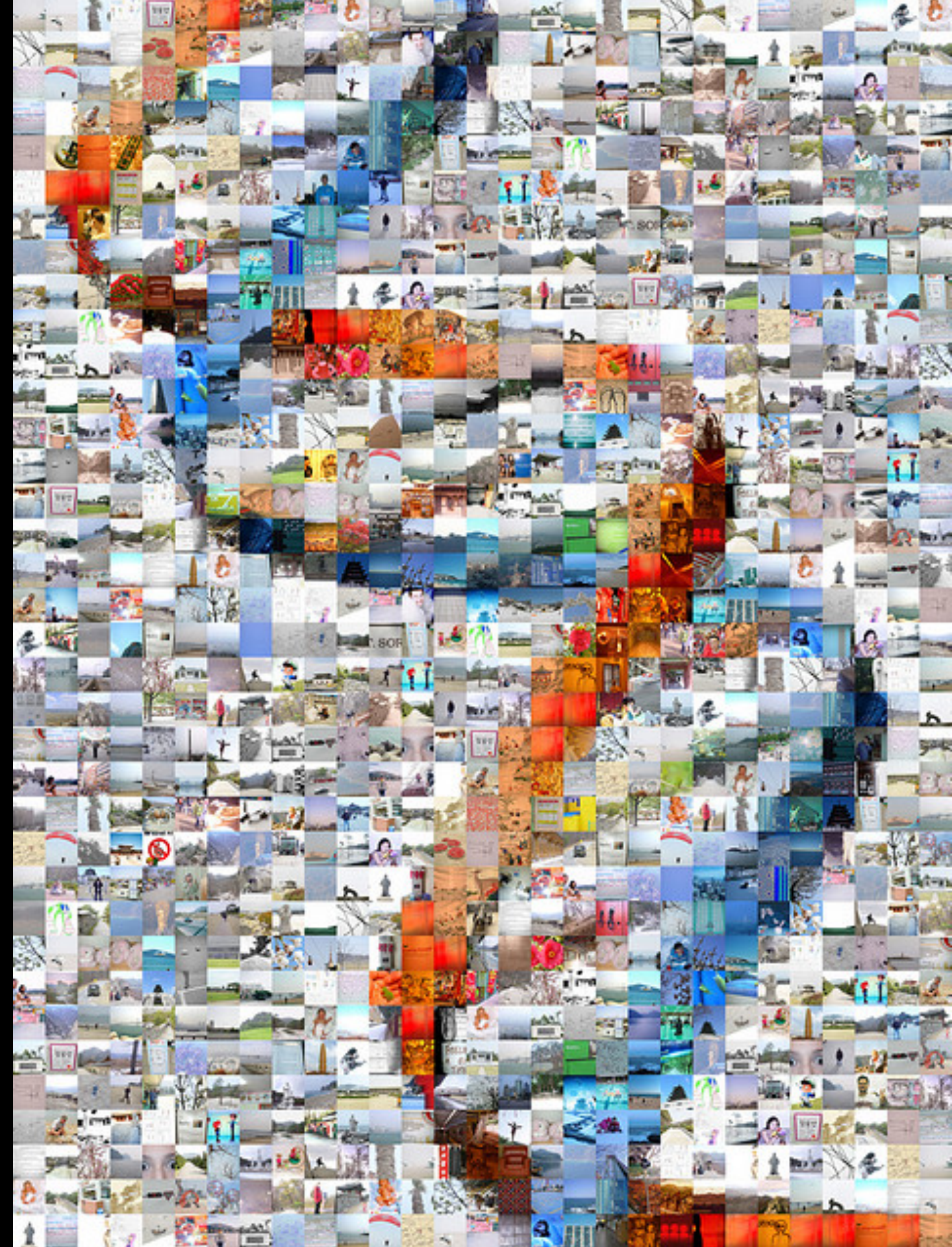
fleshed out ui layer that could support rapid experimentation

march 2012

david Marcus becomes president of PayPal

april 2012

formed lean ux team to reinvent checkout experience



Sandbox

[Home](#)[Test Accounts](#)[Test Email](#)[API Credentials](#)[Test Tools](#)

Additional resources

[Documentation](#)[PayPal Developer Network](#)[Customer Support](#)

PayPal Sandbox

API Credentials

You must have credentials to test API in the PayPal Sandbox. In most cases, you will use the test accounts identified below.

The test accounts identified below are for testing only.

Note: These credentials will not work on the live PayPal.com to go live.

Test Account

Test Account: test_131206115

API Username: test_131206115

API Password: 131206115

Signature: AFc

To download the certificate, log into the account associated with the account.

[Documentation](#)[Applications](#)[Dashboard](#)[Support](#)

// REST APIs

// Native SDKs

// Built for developers

[Get Started](#)

Looking for Sandbox? Import your test accounts to continue testing. Learn [what's new](#)

change has started working its way out

Mobile SDKs^{BETA}

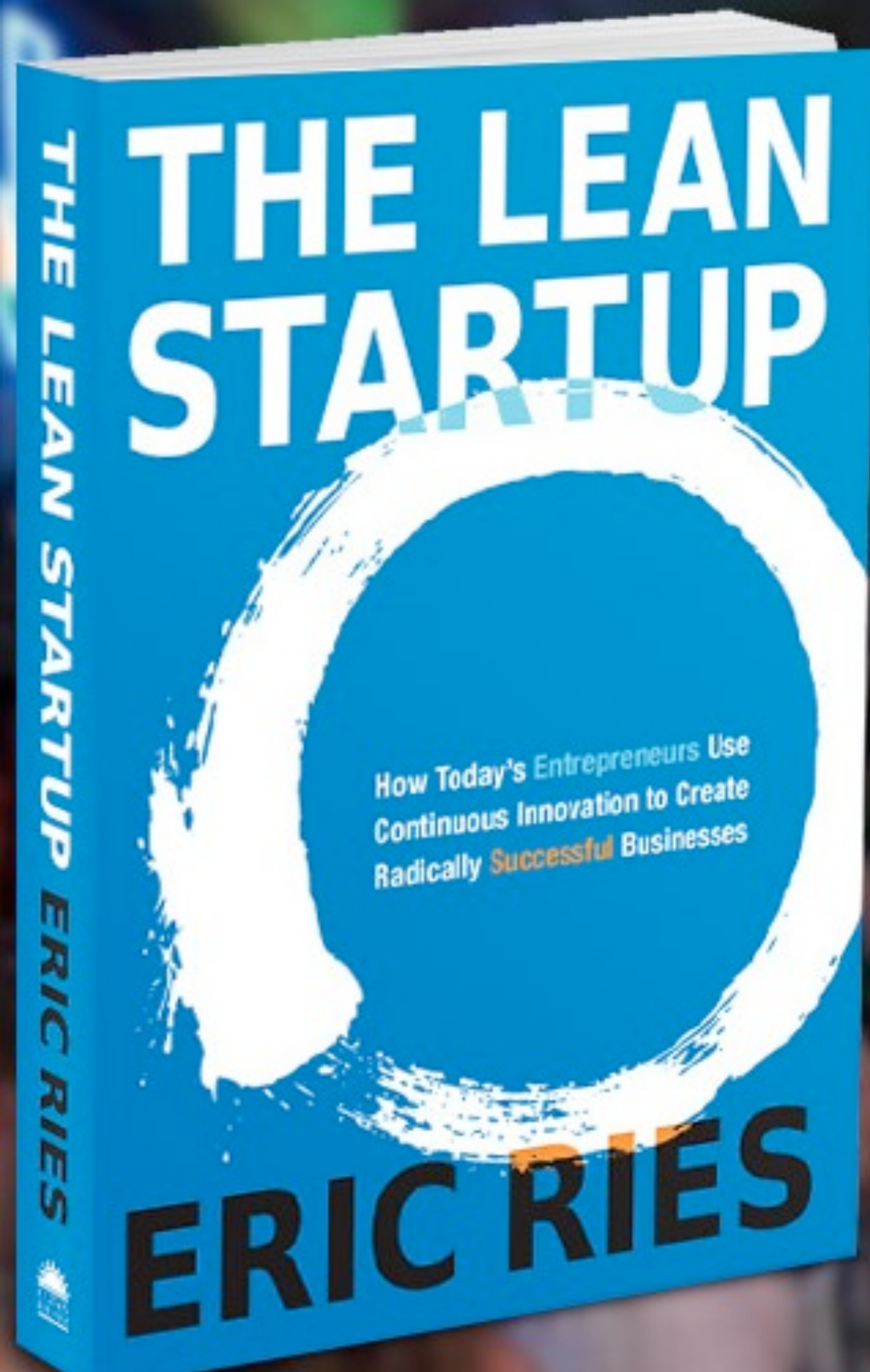
Easy, 100% native iOS SDKs that use our new REST APIs. Android coming soon.

[Download mobile SDKs](#)

REST APIs^{BETA}

Great for simple payments using PayPal or credit cards. More REST APIs on the way.

[See our REST APIs](#)



lean startup movement

founded on build/measure/learn cycle

get out of the building (GOOB)

invalidate your risky assumptions

fail fast, learn fast

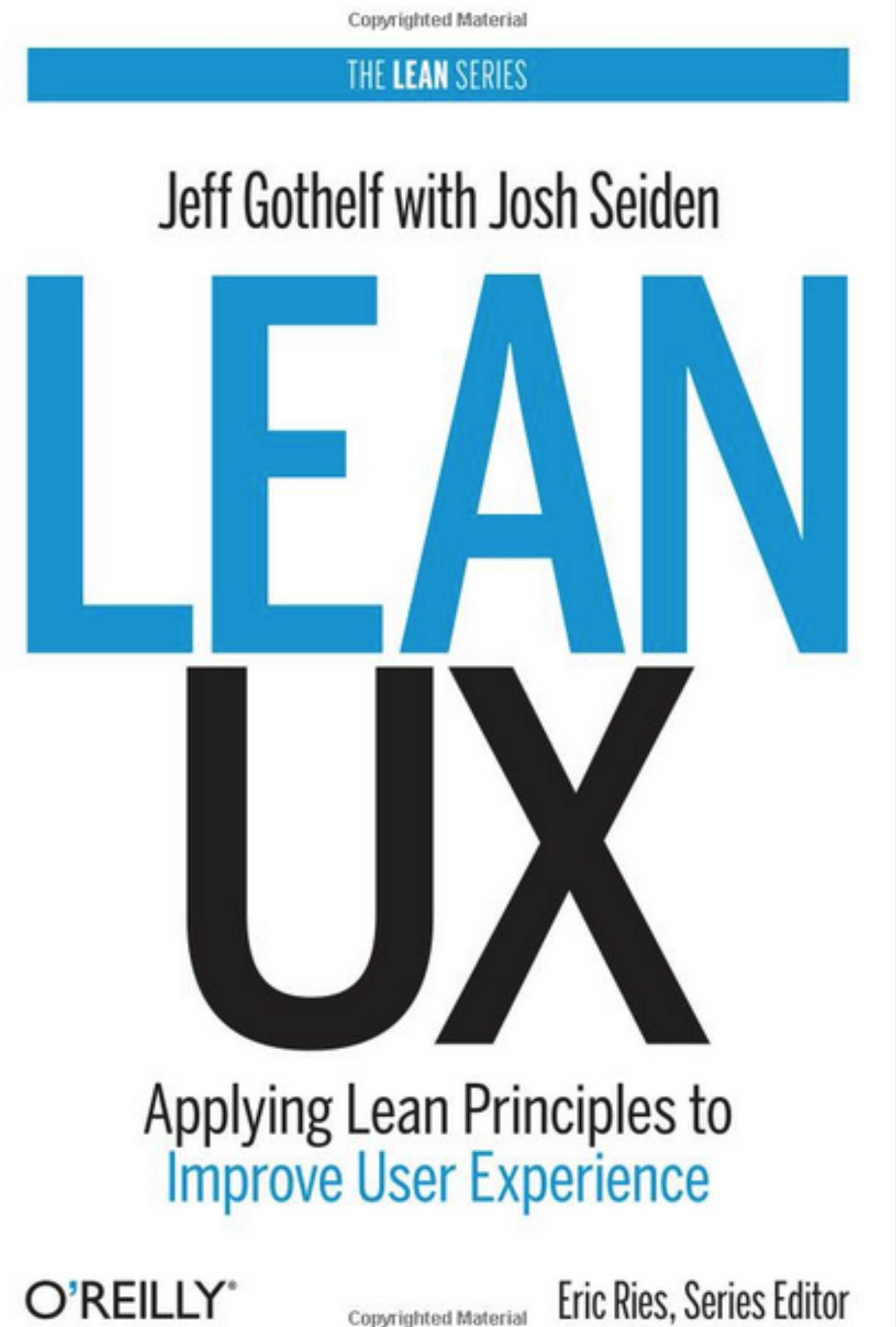
get to the pivot

go for the minimal viable product (MVP)

lean ux

designing products for build/measure/learn
requires 3 rules to be followed at all times

get to & maintain a **shared understanding**
form **deep collaboration** across disciplines
keep **continuous customer feedback** flowing



LEAN ENGINEERING

Applying Lean Startup
Principles to
Bring Design to Life

engineering driven by lean
startup principles & leanux

engineering the build/measure/learn cycle
shift the focus to minimal viable everything (MV*)

follows the key rules of lean ux:

shared understanding with design & product
deep collaboration with design & product
part of **continuous customer feedback** loop

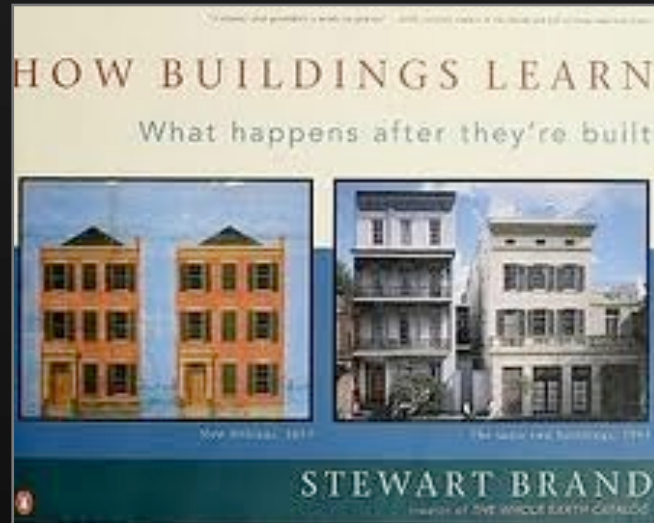
LEAN ENGINEERING

KEY LESSONS



1. engineer for experimentation

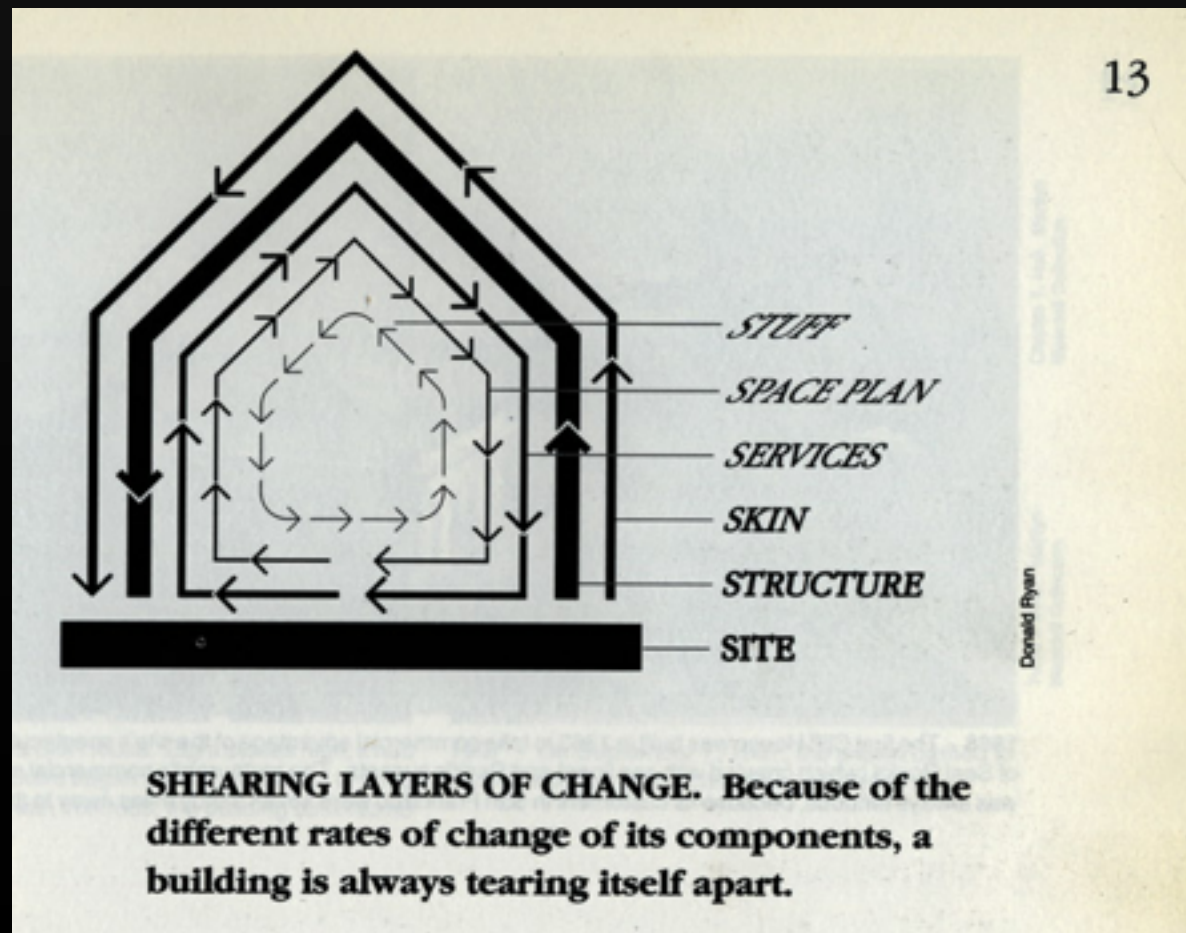
experiences must learn



All buildings are predictions.
All predictions are wrong.

There's no escape from this grim
syllogism, but it can be softened.

Stewart Brand



Our software is always tearing itself apart
(or should be)

Recognize that different layers change at
different velocities

one of our biggest challenges is moving
from a **culture of delivery** to a
culture of learning

engineering for lean ux

Lean UX - co-located



Product/Design team



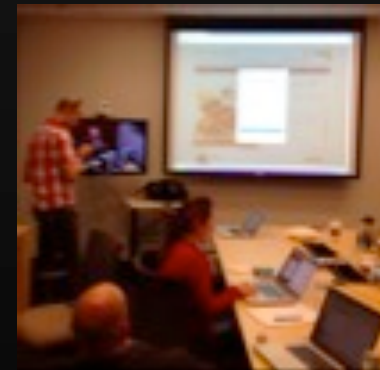
Whiteboard
to code

UIEs



Code to
Usability

Usability/Customers



And back
again...

engineering stack should

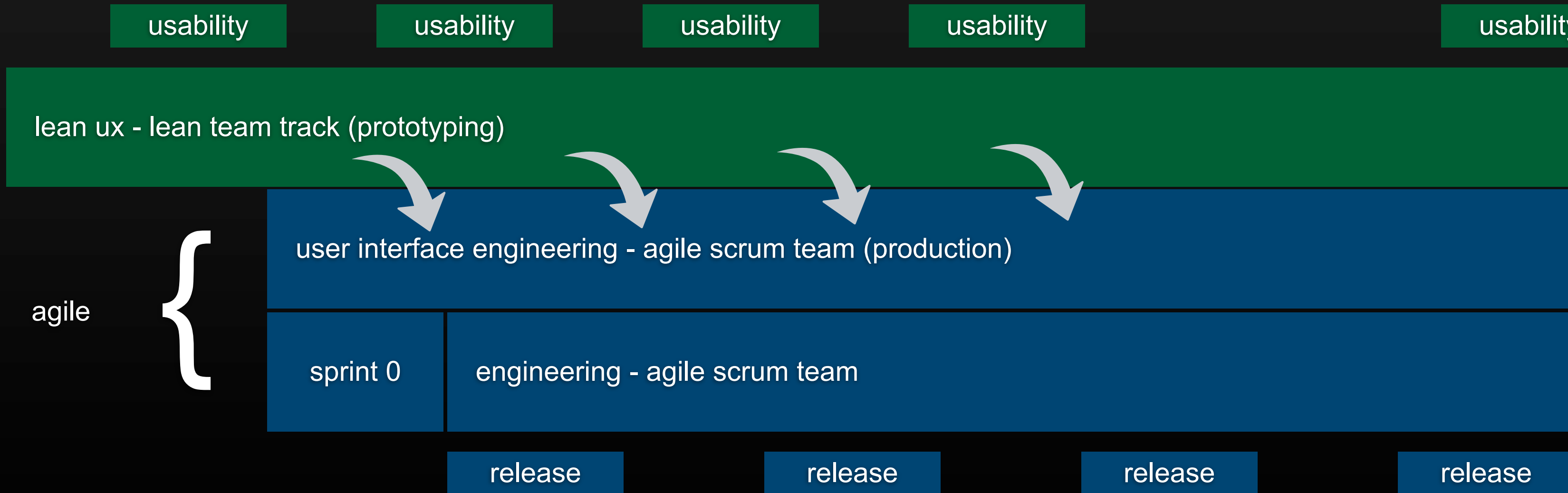
- treat prototype & production the same

- allow rapid sketch to code life cycle

- allow quick changes during usability studies (RITE)

- support being the “living spec”

lean ux: enable a brain for agile





2. refactor your way out of debt

technical debt

we have to be on modern tech stacks to
continuously innovate

we have to be on a **continuously
available** stack

continuously integrating

continuously deploying



minimal viable everything (MV*)

mvp is a key tenant of lean startup

applied to engineering we should think: minimal viable everything. mv*

minimal viable process

minimal viable team size

minimal viable technology

minimal viable tools

what are startups using?

minimal viable everything (MV*)

mvp is a key tenant of lean startup

applied to engineering we should think: minimal viable everything. mv*

minimal viable process

minimal viable team size

minimal viable technology

minimal viable tools

what are startups using?

zuck's rule: how will the startup down the street do this tomorrow? do it like that now.

stack circa 2011/early 2012

prototyping
was hard

“ui bits” could
only live here

restricted
capabilities*

client

server side
components**

server

jsp***

simple change could take minutes
to see

java

follows an “enterprise application”
model. ui gets built into the “app”

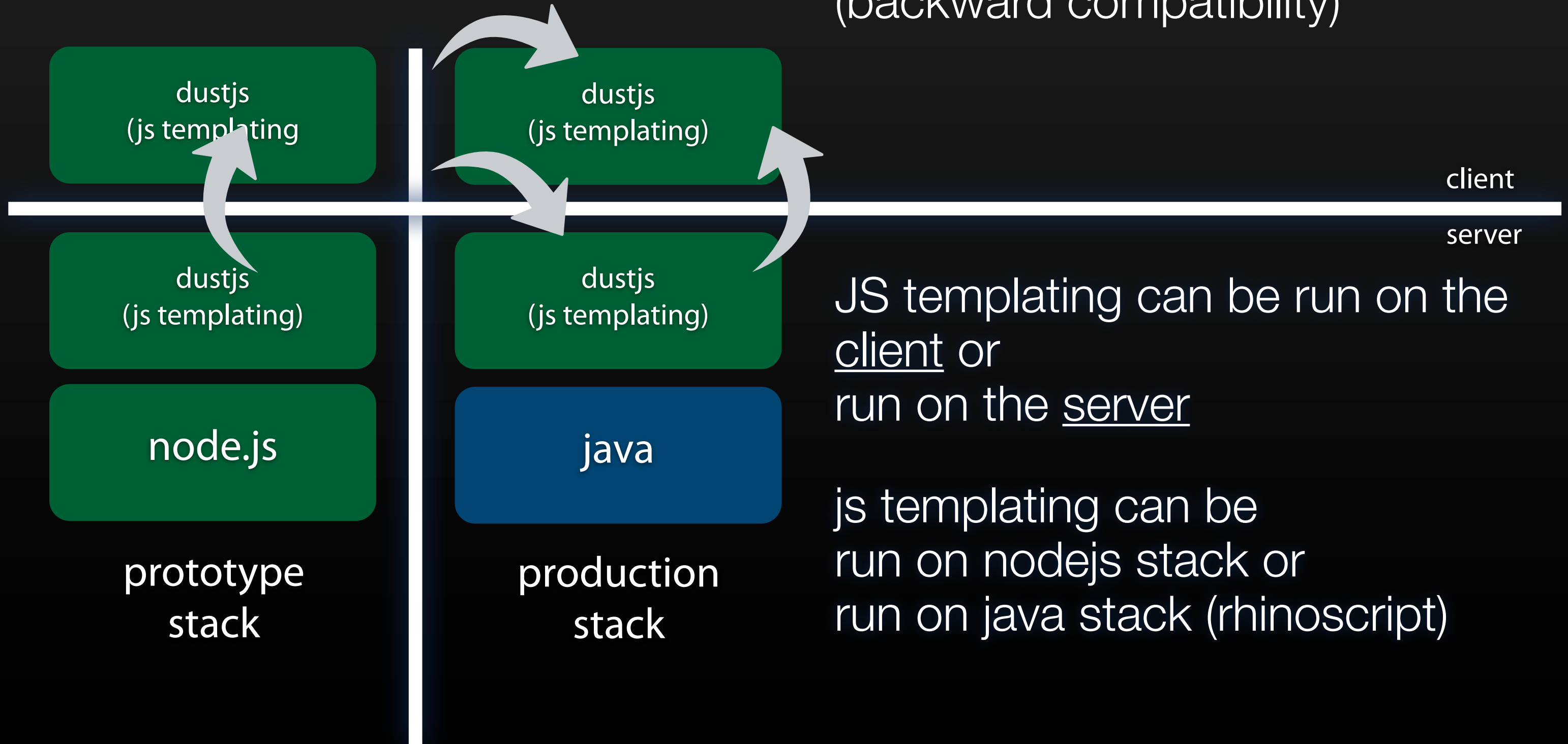
* assumed client developers were low-skill

* required server side java eng for simple client changes

** java server pages. server-side java templating solution

ui portability

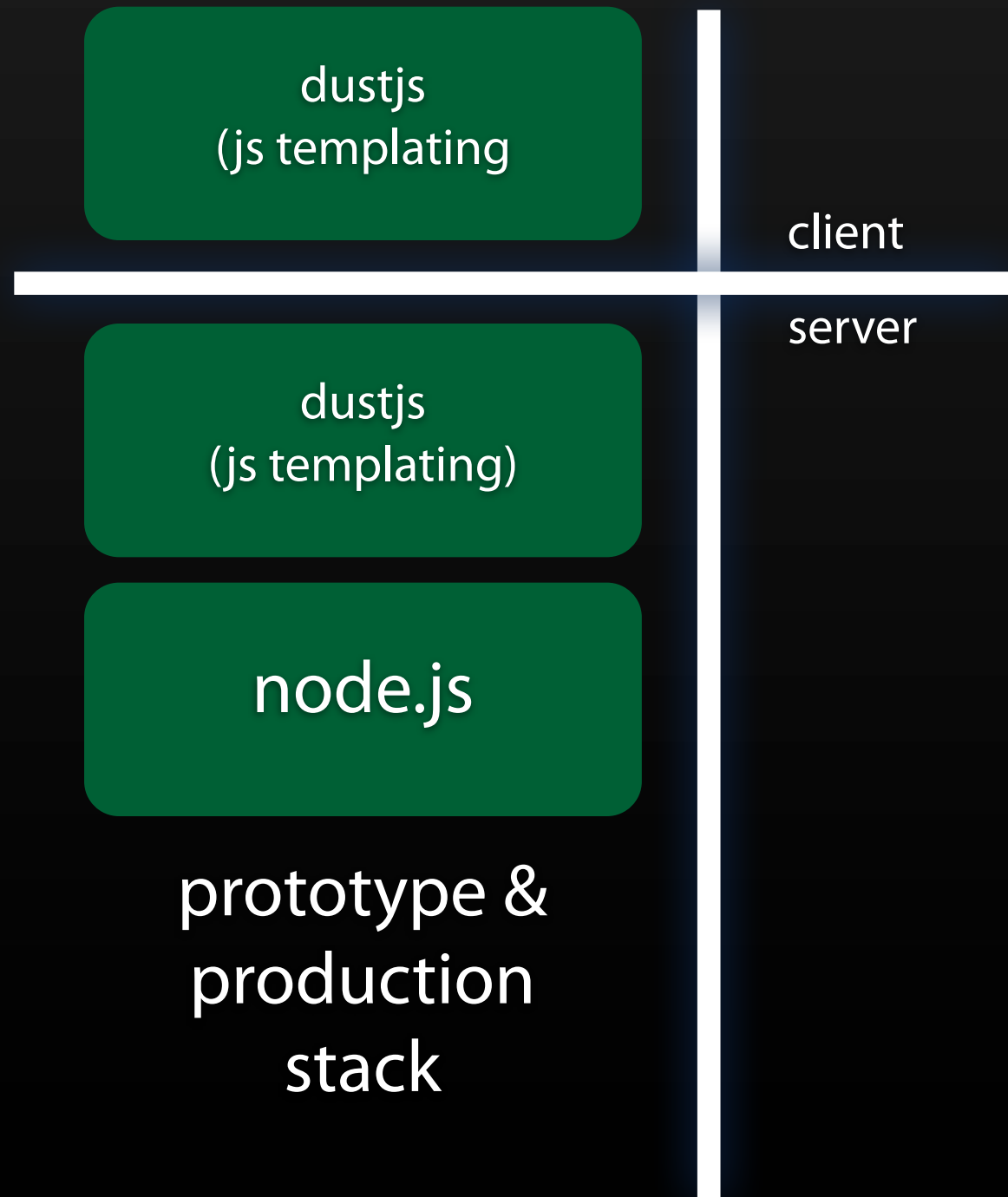
allowed us to refactor
to get to the new stack
(backward compatibility)



JS templating can be run on the
client or
run on the server

js templating can be
run on nodejs stack or
run on java stack (rhinoscript)

one stack. prototype or production.





bad design kills

experience debt

don't just think about our technical debt
consider our "experience debt"
cripples our ability to capture market and
inhibits learning

github

SOCIAL CODING



3. be open source top to bottom

use open source religiously

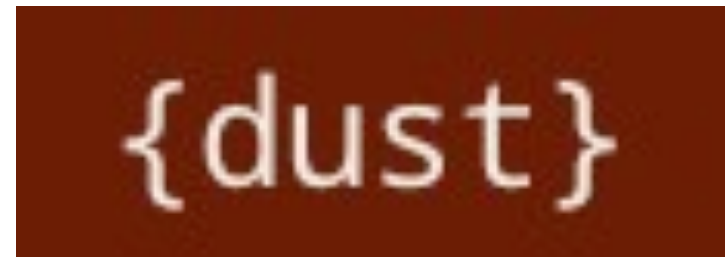
Bootstrap, from Twitter

express



BACKBONE.JS

UNDERSCORE.JS



BOWER

work in open source model

internal github revolutionizing
our internal development

rapidly replacing centralized
platform teams

innovation democratized

every developer encouraged
to experiment and generate repos
to share as well as to fork/pull request



give back to open source

we have projects that we will open source

- node webcore (similar to yeoman)

we are contributing back to open source

- contributions to bootstrap (for accessibility)

- contributions to bootstrap (for internationalization)

- core committer on dustjs project

using github for continuous *

use github for continuous integration

starting to use github repo model for continuous deployment

- marketing pages

- product pages

- content updates & triggers into i18n, l10n, adaptation

- components

summary

to support collaboration, shared understanding and continuous customer feedback we are changing:

technology. but not for tech sake. we are doing it for the experience to support lean startup principles.

process. enabled lean ux and put a brain on agile.

people. revitalizing our existing talent and started attracting new talent.



blogs

<http://looksgoodworkswell.com>

<http://designingwebinterfaces.com>

twitter

follow me on twitter @billwscott 

book



designing web interfaces
O'Reilly

picture credits

<http://www.flickr.com/photos/paolomargari/3793121065/sizes/z/>

http://www.flickr.com/photos/smb_flickr/439040132/

<http://www.flickr.com/photos/juanpol/16287486/sizes/z/>

<http://www.flickr.com/photos/giesenbauer/4092794246/sizes/l/>

<http://www.flickr.com/photos/therevsteve/3104267109/sizes/o/>

http://www.flickr.com/photos/not_wise/182849352/sizes/l/

<http://www.flickr.com/photos/37217398@N02/3442676067/sizes/l/>

<http://www.flickr.com/photos/hongiiiv/4151964823/sizes/z/>

<http://www.flickr.com/photos/therevsteve/3104267109/sizes/o/>

<http://www.flickr.com/photos/mbiskoping/6075387388/>

<http://www.flickr.com/photos/soldiersmediacenter/4685688778/sizes/z/>

<http://www.flickr.com/photos/epsos/8463683689/sizes/l/>

<http://www.flickr.com/photos/proimos/3473264448/sizes/l/>

<http://www.flickr.com/photos/54459164@N00/6452577109/sizes/l/>