

Building Pinterest's Mobile Apps



Mike Beltzner
Product Manager

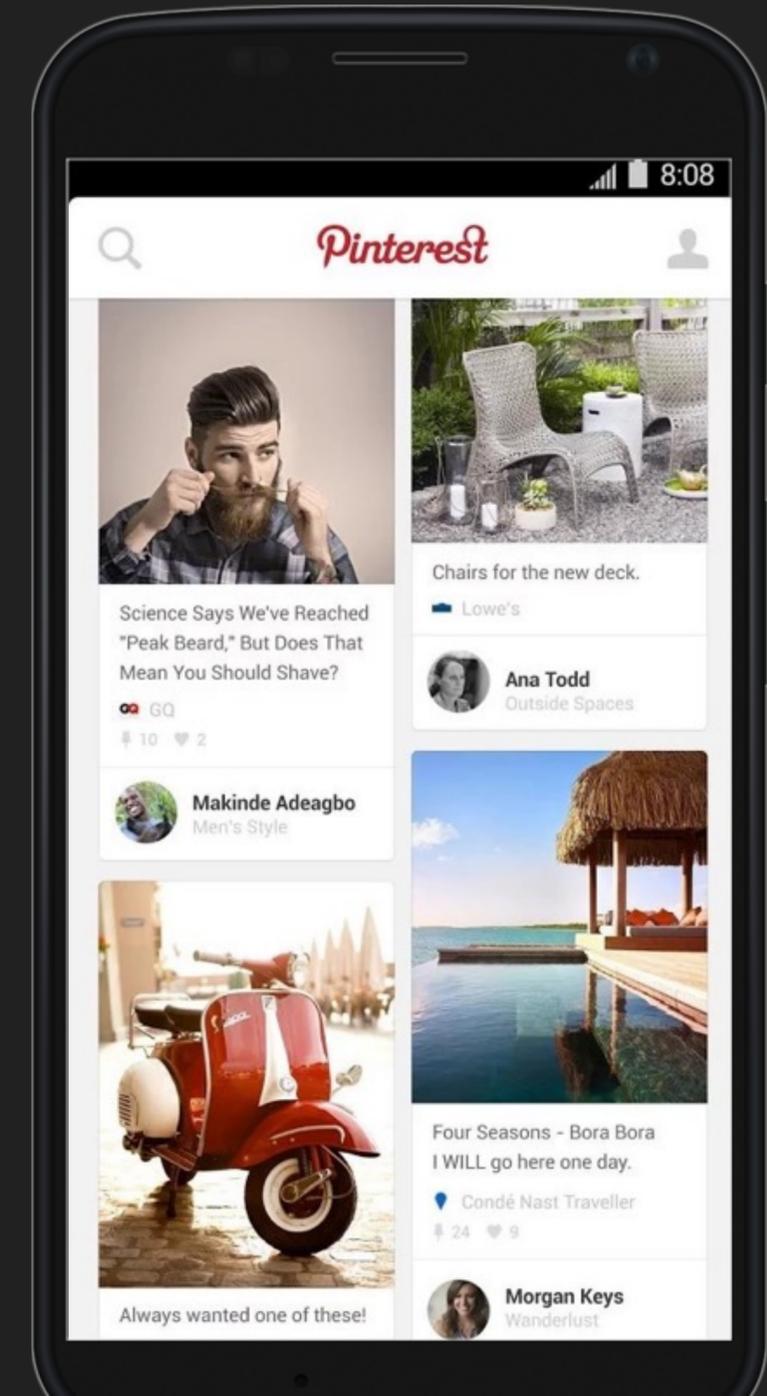
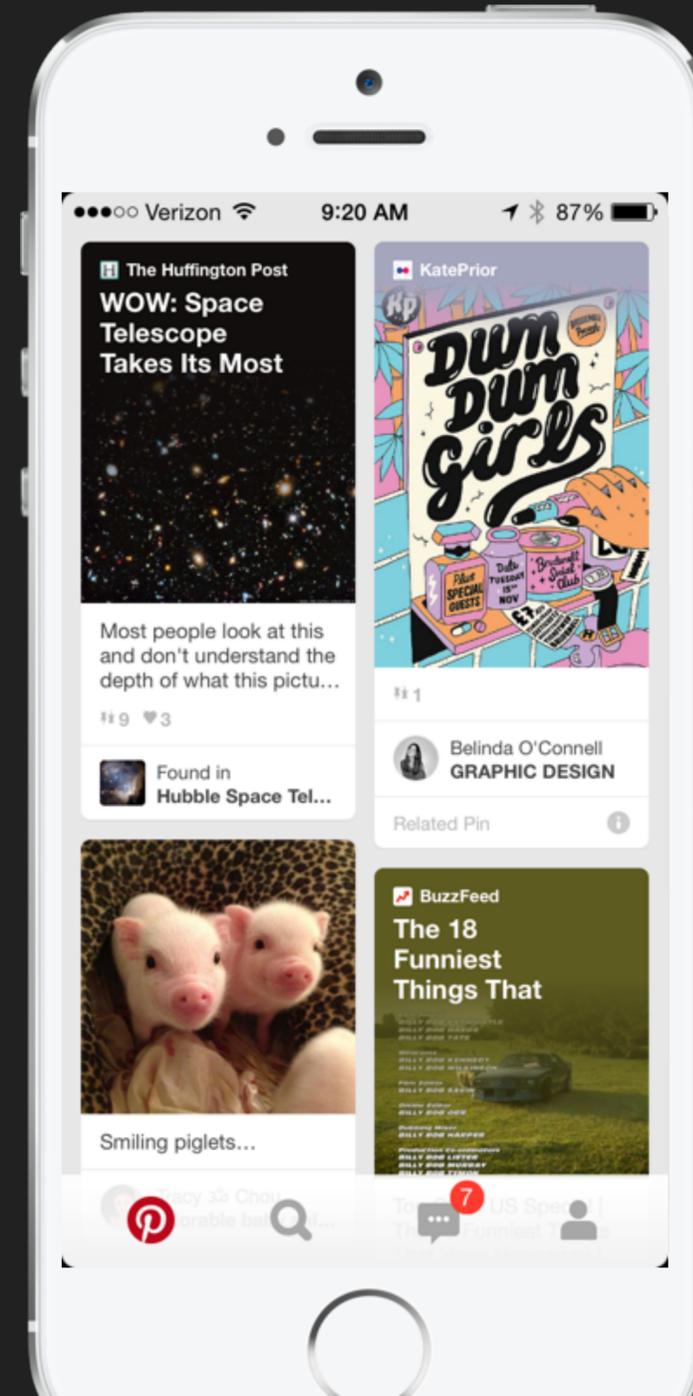


Garrett Moon
Mobile Team (iOS)



Pinterest and mobile

- Launched iPhone app in 2011
- “Summer of Apps” in 2012
- 75% of usage on Pinterest is mobile
 - iOS highest by volume
 - Android fastest growing
- Will ship 16 times/yr/platform
 - Not including support releases
 - Fewer than 0.5% sessions crash
 - Maintained 4.5+ star rating



“Uh oh”

PRODUCT MANAGER

#partylikeaproductmanager



What my friends think I do



What my mom thinks I do



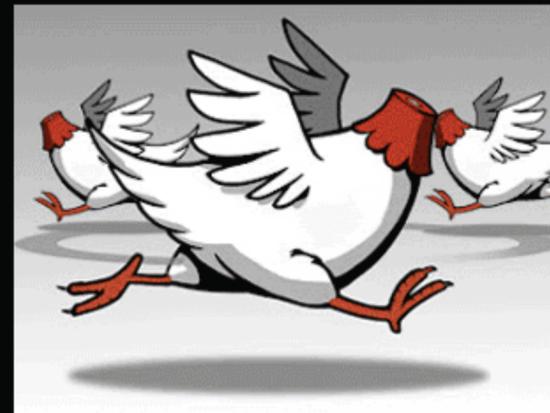
What society thinks I do



What my colleagues think I do



What I think I do



What I actually do

#TBT



2000

- Creative Nomad
- Command and Conquer: Red Alert 2
- Internet Explorer 5.0

- Software sold like records, in stores
 - long production cycles
 - QA validation
 - “GM”, “Hotfix”, sales data & CSAT
 - shelf location matters



2006

- iPod Nano
- World of Warcraft
- Firefox 1.0
- Software was “web-first”
 - Continuous deployment
 - “Move fast and break things”
 - A/B testing
 - search rank matters

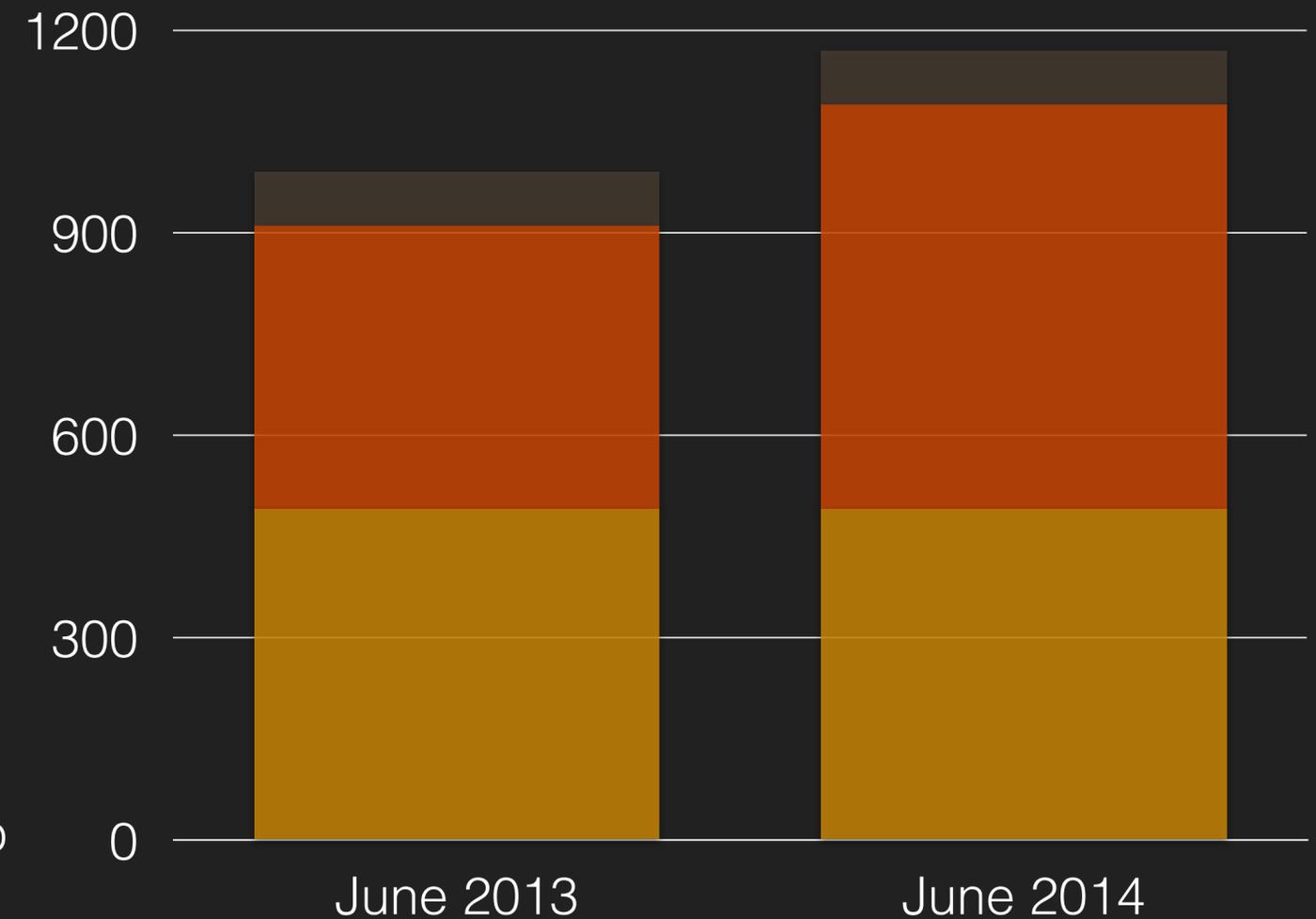


2008



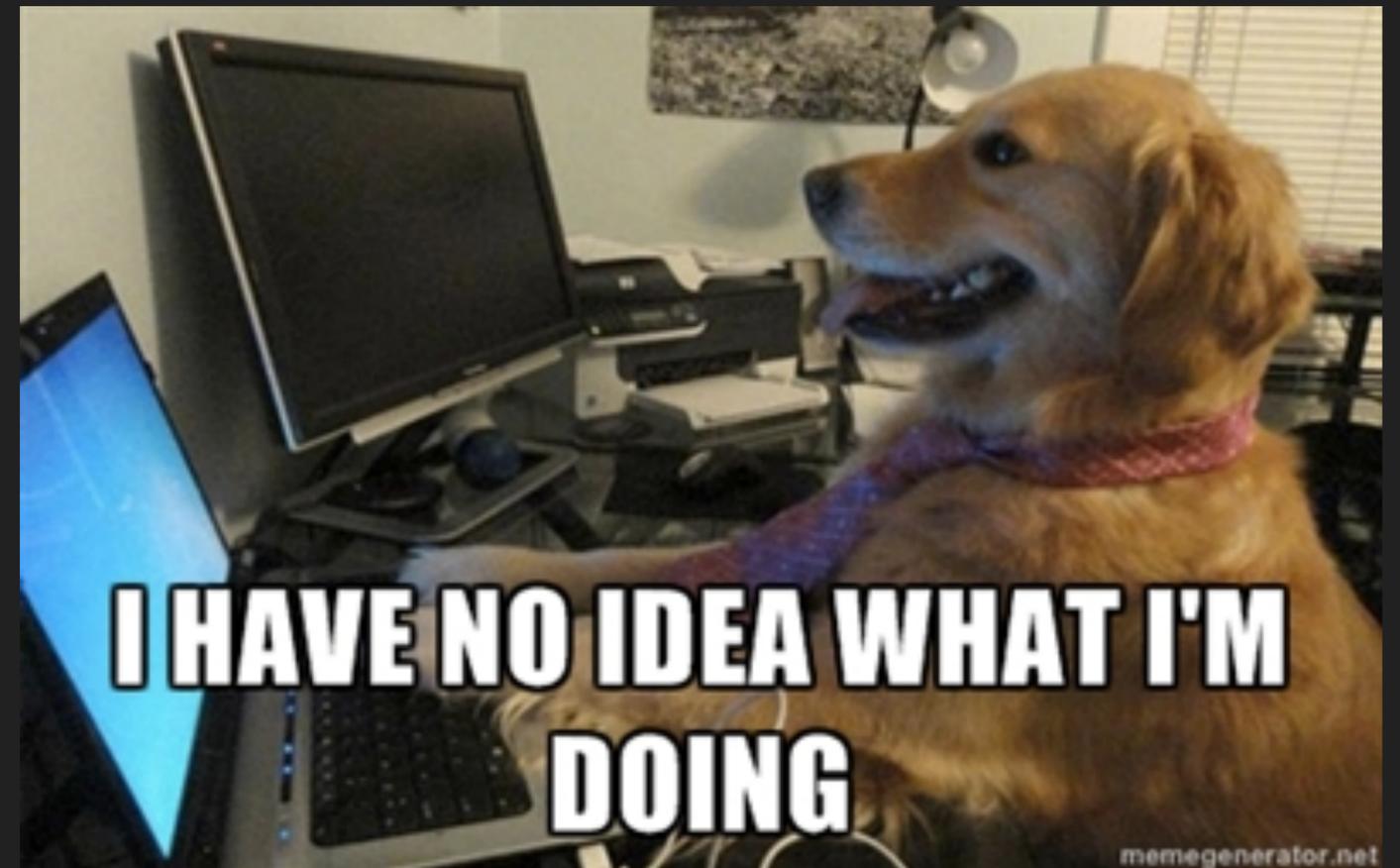
2014

- “Mobile is eating the world”
- 3B people online; 2B w/smartphones
- More time spent in mobile apps than on all of the web



Apps are different

- Not packaged software
- Not web software
- We're still learning how to do it
 - planning
 - execution
 - deployment
 - move fast ... but don't break things



Lessons from 14 years ago

- Validate work with thorough testing
- Understand to predict causality
- Monitor output metrics
- Relationships lead to distribution
- Measure twice, cut once



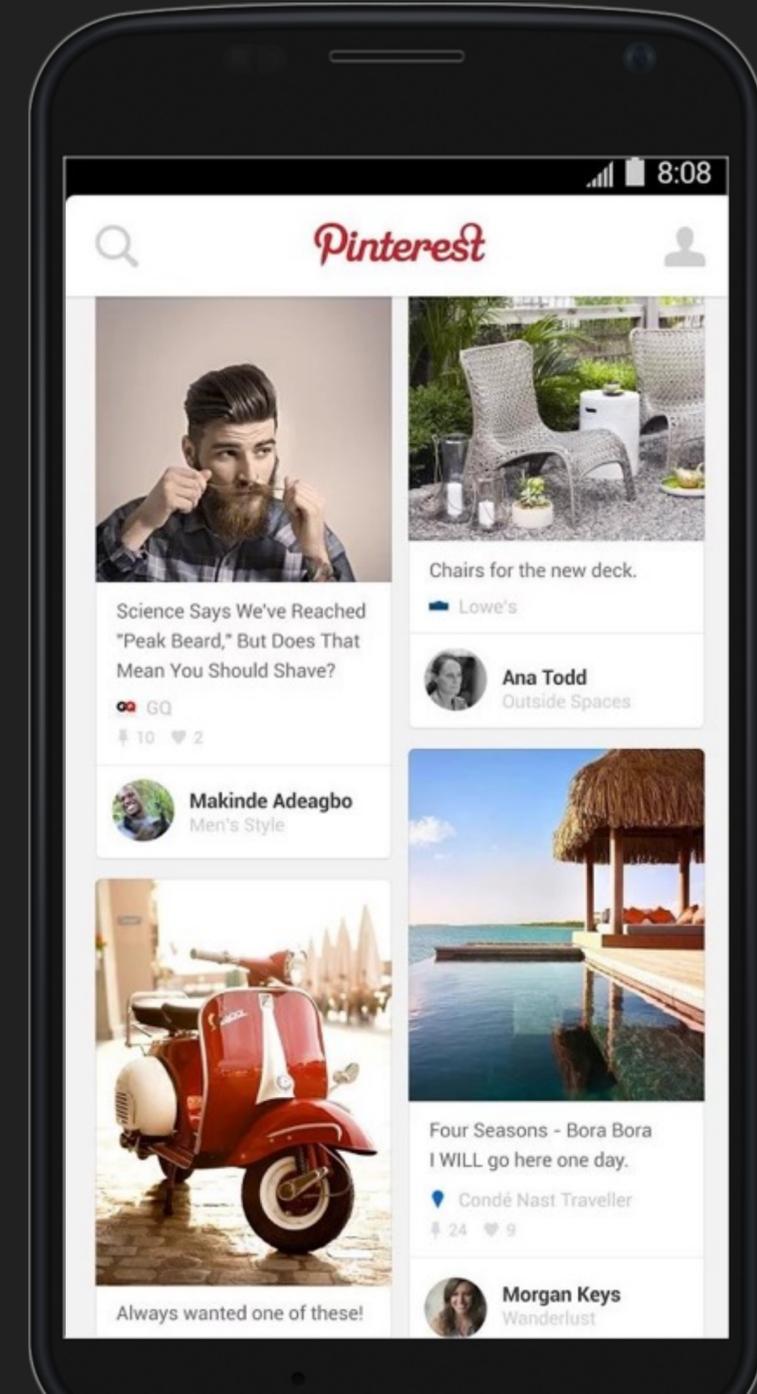
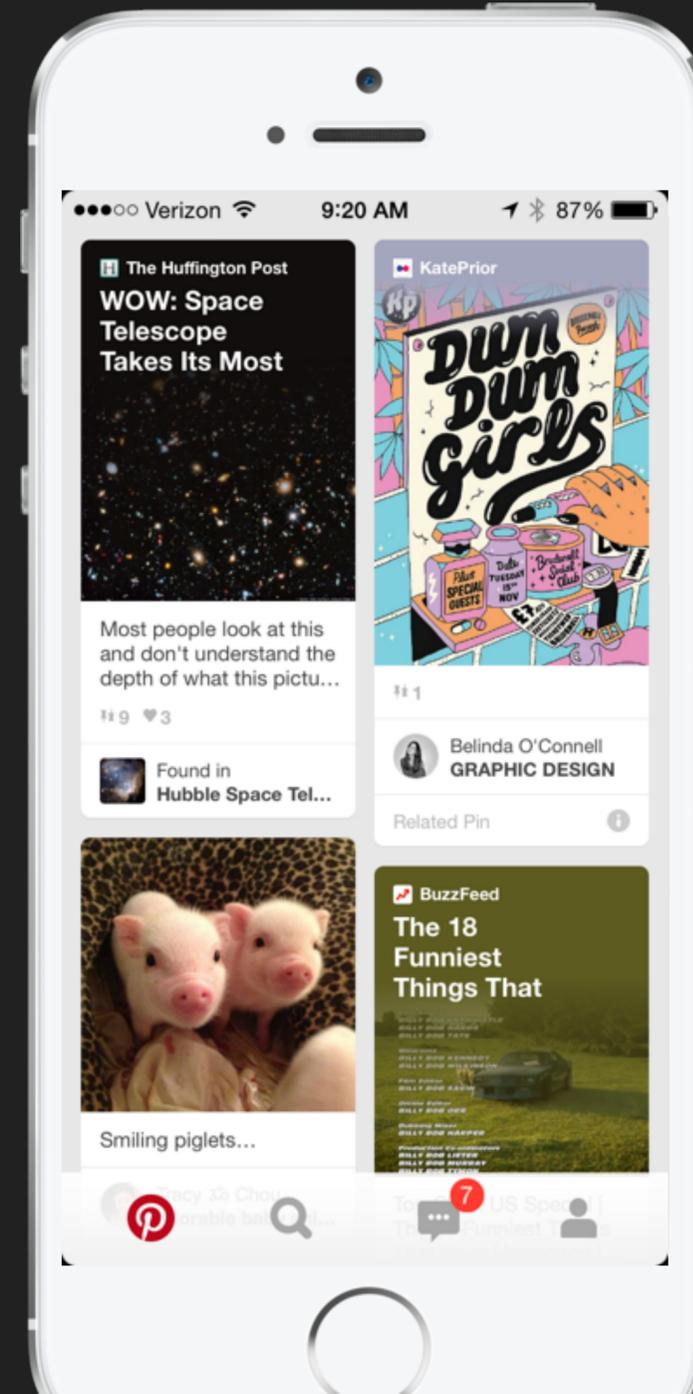
Lessons from 7 years ago

- Bugs most readily found by usage
- Experiment to understand causality
- Monitor everything
- SEO leads to distribution
- Ship first, ask later



Lessons applied

- Prevent big bugs from shipping
- Form and validate hypotheses
- Log & monitor everything
- SEO for app upsell
- Relationships for initial exposure
- Ship frequently and with regularity



Prevent bugs from shipping

- Automated testing
 - on every build
 - before it gets merged
 - unit & integration
- Human testing
 - dogfooding
 - daily integration testing
 - lightweight “certification”
- Limited rollout (fragmentation)
 - Google Play
 - TestFlight

Experimenting

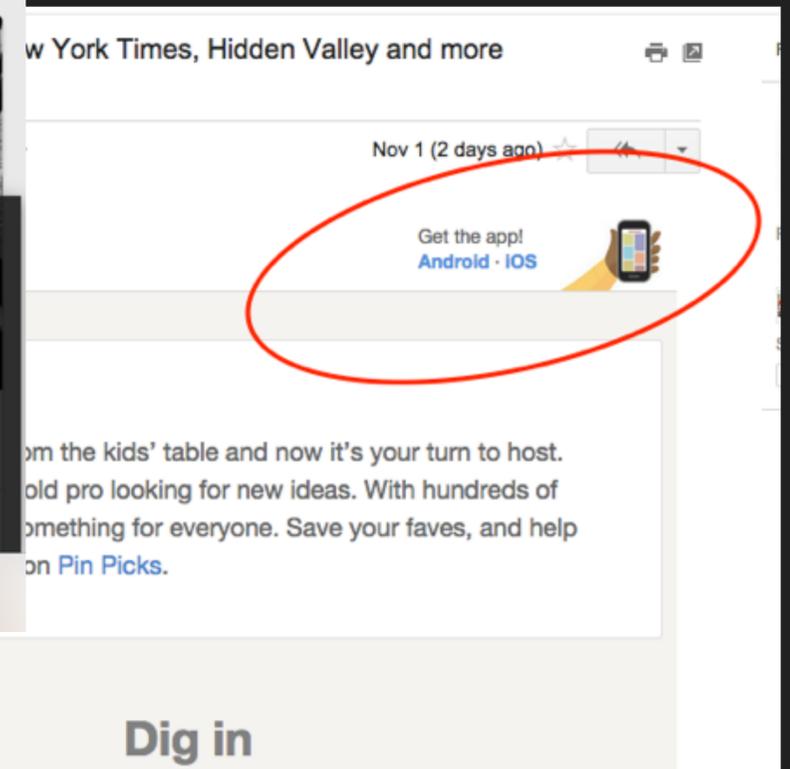
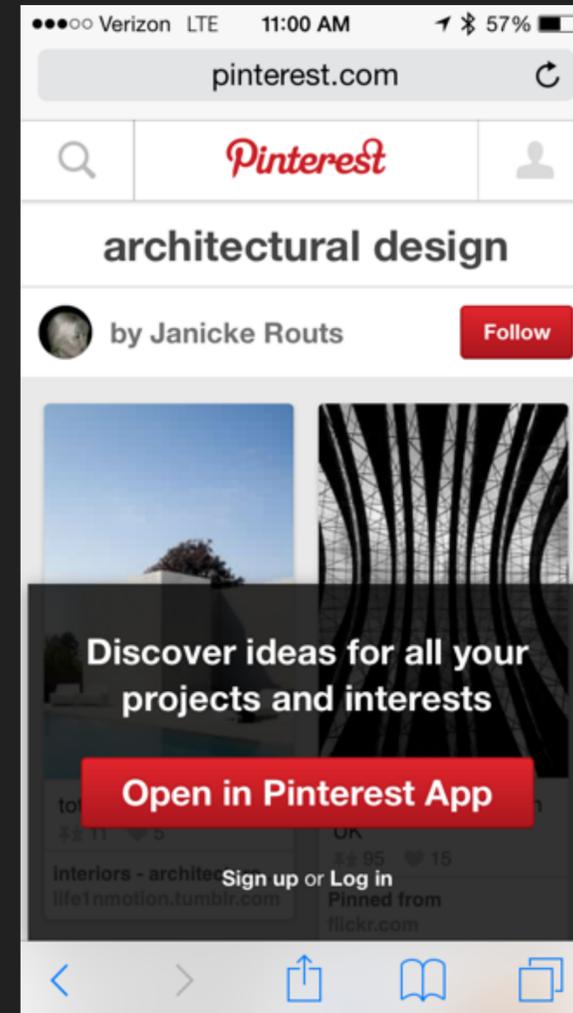
- Everything is an experiment
 - to learn & explore
 - to understand effects
- Short and long term analyses
- You are the only one desperate for your new feature

Monitoring

- Log everything
 - API requests
 - client side usage
- Monitor to spot problems
 - Crash rates
 - Usage metrics
 - App Store ratings (iRate)
 - App Store conversions

SEO and upsell

- SEO / web good for distribution
- Best interstitials demonstrate value
- Always be upselling to mobile



Relationships matter

- Understand how the store works
 - country & region managers
 - give & take on technology development
- New technologies offer opportunity
 - App Indexing
 - App Links

Ship Frequently & Regularly

- 3 week “release trains”
- large projects go over multiple cycles
- single branch prevents integration hell

Ship Frequently & Regularly

Mon	Tue	Wed	Thu	Fri
			START	

Ship Frequently & Regularly

Mon	Tue	Wed	Thu	Fri
			START	CODE
CODE	CODE	CODE	CODE	CODE
CODE	CODE	CODE	CODE	FREEZE
BUGS	BUGS	SUBMIT		

Ship Frequently & Regularly

Mon	Tue	Wed	Thu	Fri
			START	CODE
CODE	CODE	CODE	CODE	CODE
CODE	CODE	CODE	CODE	FREEZE
BUGS	BUGS	SUBMIT	WAIT	WAIT
WAIT	WAIT	WAIT	SHIP	

Ship Frequently & Regularly

Mon	Tue	Wed	Thu	Fri
			START	CODE
CODE	CODE	CODE	CODE	CODE
CODE	CODE	CODE	CODE	FREEZE
BUGS	BUGS	SUBMIT	START	CODE
CODE	CODE	CODE	SHIP	CODE
CODE	CODE	CODE	CODE	FREEZE

Ship Frequently & Regularly

Mon	Tue	Wed	Thu	Fri
BUGS	BUGS	SUBMIT	START	CODE
CODE	CODE	CODE	SHIP	CODE
CODE	CODE	CODE	CODE	FREEZE
BUGS	BUGS	SUBMIT	START	CODE
CODE	CODE	CODE	SHIP	CODE
CODE	CODE	CODE	CODE	FREEZE

Ship Frequently & Regularly

Mon	Tue	Wed	Thu	Fri
BUGS	BUGS	SUBMIT	START	CODE
CODE	CODE	CODE	SHIP	CODE
CODE	CODE	CODE	CODE	FREEZE
BUGS	BUGS	SUBMIT	START	CODE
CODE	CODE	CODE	SHIP	CODE
CODE	CODE	CODE	CODE	FREEZE
BUGS	BUGS	SUBMIT	START	CODE
CODE	CODE	CODE	SHIP	CODE
CODE	CODE	CODE	CODE	FREEZE
BUGS	BUGS	SUBMIT		

Ship Frequently & Regularly

- predictability helps all teams plan
- allows us to budget platform vs feature
- took us about 3 release cycles
- *reduces* the need for hot fixes
 - forces you to focus on stability
 - changes the cost of a bug

November 2014							< Today >
	Sun	Mon	Tue	Wed	Thu	Fri	Sat
45	2	3	4	5	6	7 4.1/iOS: Integration...	8
46	9	10	11 4.1/iOS: Submit to...	12	13 3.7/Android: Integra...	14	15
47	16	17	18 3.7/Android: 5% Ro...	19 SHIP: 4.1/iOS 3.6/...	20	21	22
48	23	24	25 4.1.1/iOS: Integratio...	26	27	28	29
49	30	Dec 1	2 4.1.1/iOS: Submit t...	3	4	5 3.8/Android: Integra...	6
50	7	8	9 3.8/Android: 5% Ro...	10 SHIP: 4.1.1/iOS 3...	11	12	13

The future of mobile? Mobile is the future.

- 75% of Pinterest's usage is mobile.
- Mobile First.
- Mobile Best.

Engineering for your most important platforms

- Integrate deeply with your platform
- Increase performance
- Reduce crash rates
- Eliminate hangs
- Know when to rely on others and when to write your own code

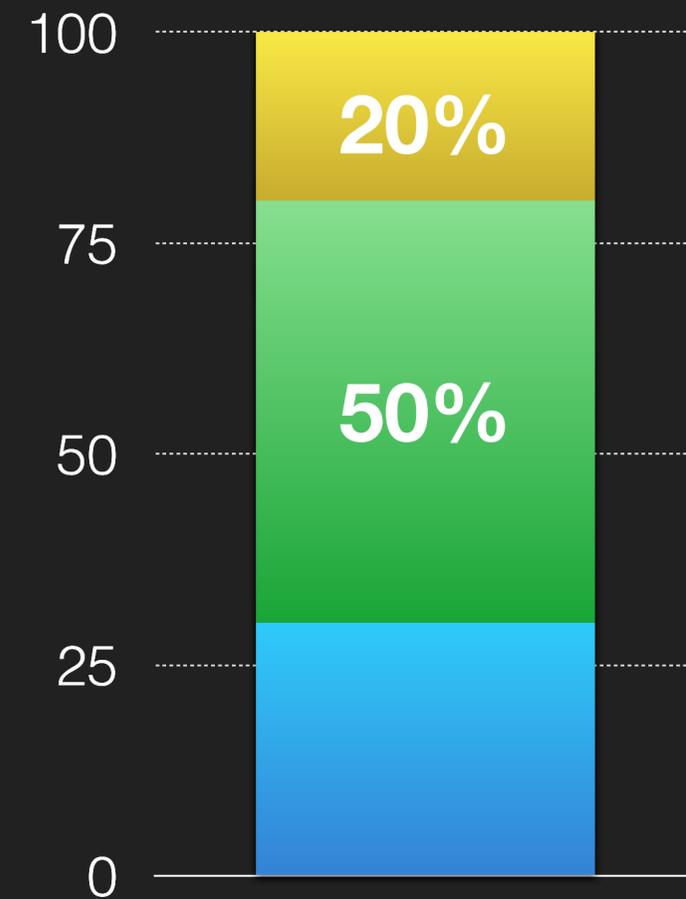
Platform integration

- Share extensions ([App extension programming guide](#))
- Safari Keychain integration ([SecRequestSharedWebCredential](#))
- Today Center widgets



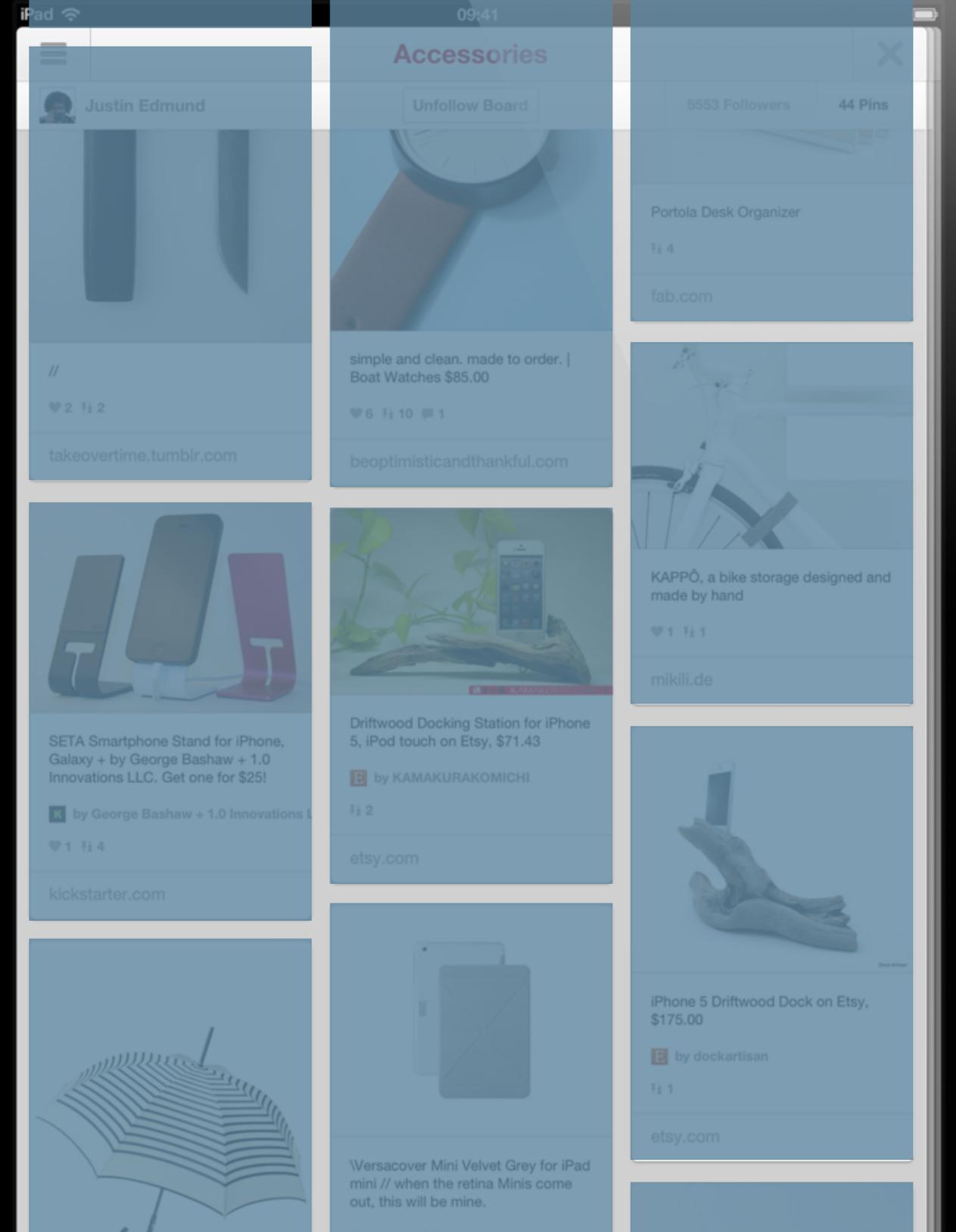
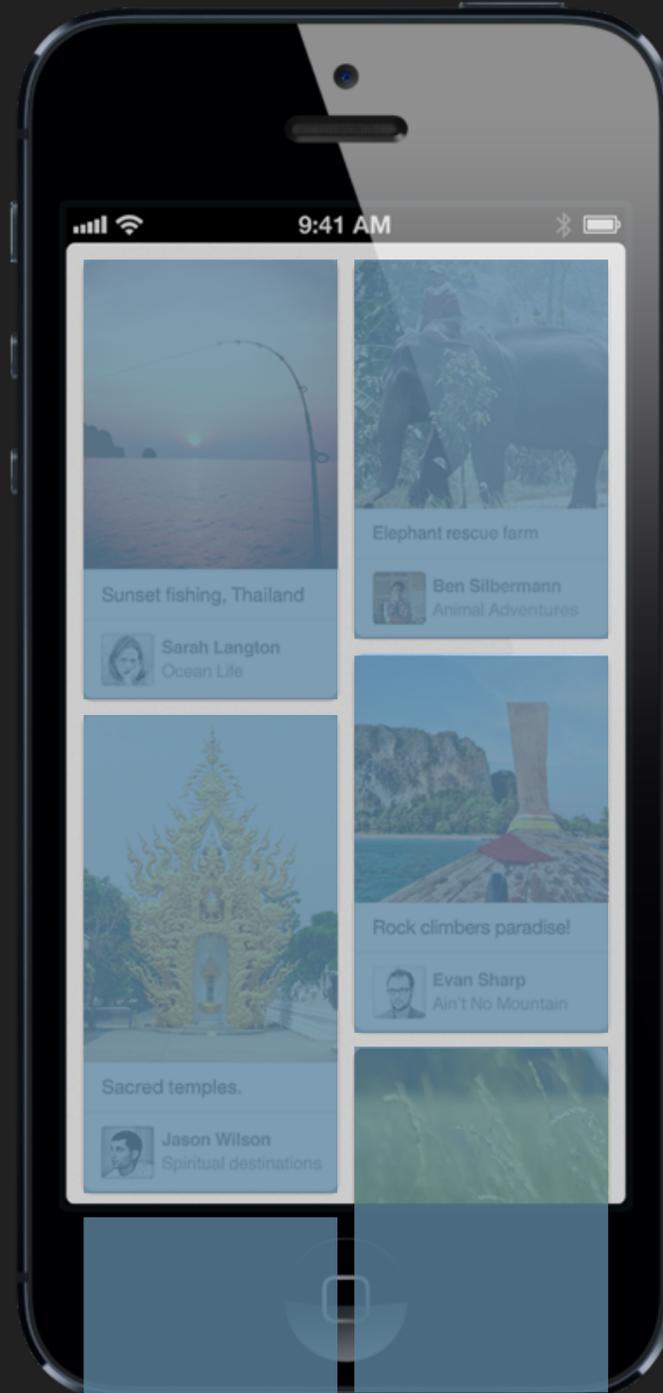
Startup Performance

- Reduce app size
- Audit app startup
- Use Instruments
- Defer if possible
- Use `dispatch_async`
- Cache
- Be careful of rogue third party libraries



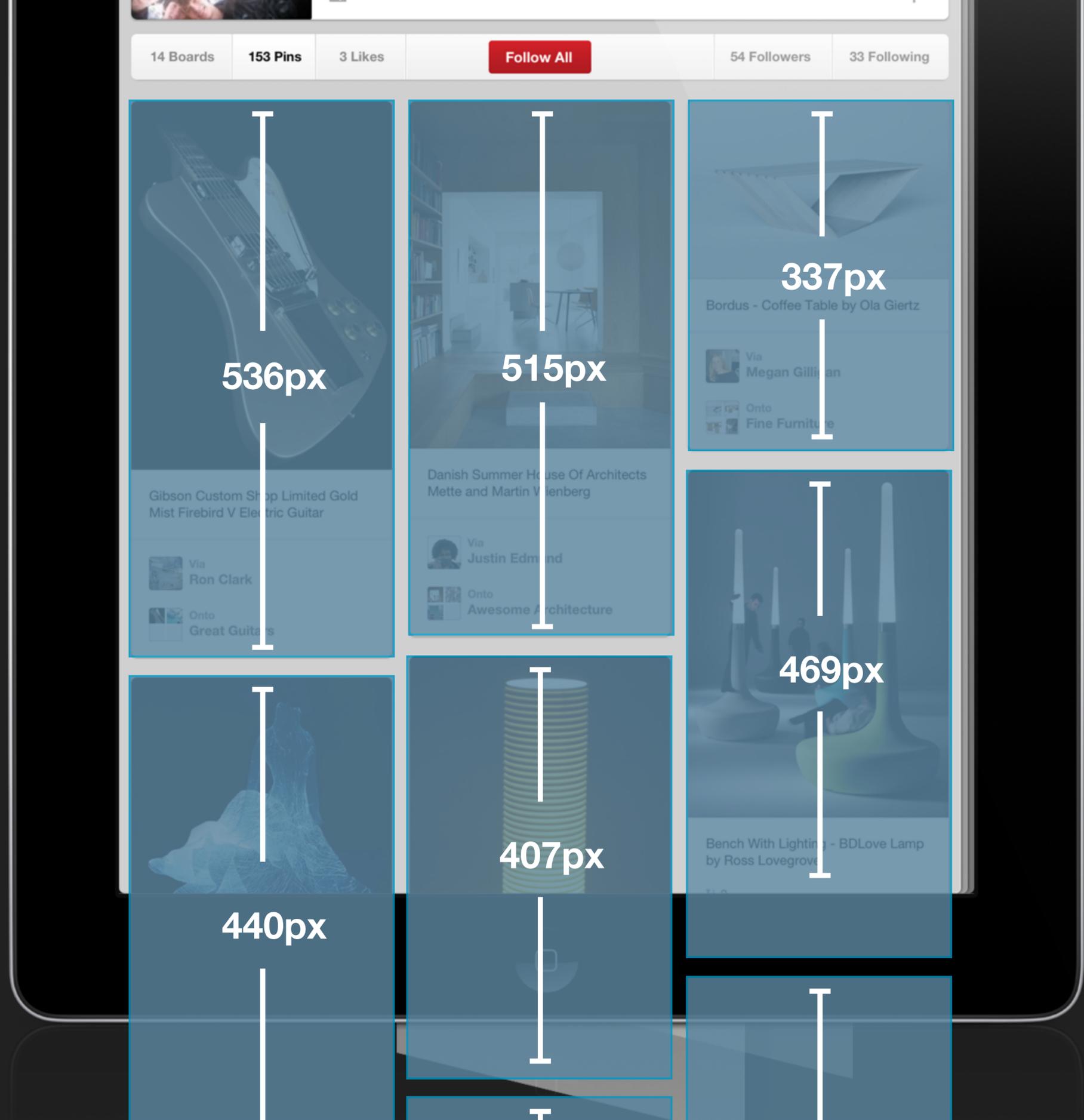
Scrolling

- Size images appropriately
- Decompress images off main thread
 - UIImage decompresses on demand
- UIImage drawing methods are thread safe



Scrolling

- Optimized shadows
- Flattened cells
- Non transparent views
- Cell height caching



Why you need better crash reporting

- Users affected
- Sessions affected
- Devices affected

Eliminate hangs

- Often deadlocks
 - Possibly stuck in a loop
 - Probably blocked waiting on another thread
- Sometimes corrupted UI

Lean on others

- CocoaPods
- AFNetworking
- Smartling
- FLAnimatedImage
- TMCache

The logo for CocoaPods, featuring the text '<COCOAPODS>' in a bold, red, sans-serif font. The text is centered within a white rounded rectangular box.The logo for AFNetworking, featuring a stylized orange and yellow flame icon above the text 'AFNETWORKING' in a bold, black, sans-serif font. The text is centered within a white rounded rectangular box.The logo for Smartling, featuring the text 'smartling' in a lowercase, grey, sans-serif font, followed by a small orange speech bubble icon containing the text 'HI!'. The text is centered within a white rounded rectangular box.

When to reinvent the wheel

- Image downloading and caching
 - Some aren't performant
 - Some are focused on performance in very specific scenarios
 - Some deadlocked when starved for threads
- What have we built?
 - Based on existing open source libraries
 - Small
 - Efficient
 - Supports GIFs natively

Thanks!
Questions?



beltzner@pinterest.com

pinterest.com/beltzner

@beltzner



garrett@pinterest.com

pinterest.com/garrettlunar

@garrettmoon

