

The Bad Habits of Mobile Software Engineers

Chris Haseman

Bad Ideas (all of them mine)

- ▶ Mobile devices are always connected
- ▶ “Simple” is just a matter of doing less
- ▶ Customize *everything* - Customize *nothing*
- ▶ Build it and they will come (to the markets)

Why should you care?



Why should you Listen?

I've f*cked up more than you have

"Phones are
Always
Connected!"

The Myth

...phones are not always connected

- ▶ The 'bars' in your status bar are theoretical
- ▶ Phone tells you what it believes could be true
- ▶ You don't know until you try to connect...
- ▶ A true 'always on' connection would eat your battery in hours
- ▶ "Add a loading spinner!" "Put in a progress bar!"

I've got some bad news...

**Now that you know about
it, you're on the hook to
help maintain it**

...there are no special jackets

why should you join the club?

- ▶ Waiting in mobile applications is particularly painful
- ▶ Failures are inevitable, dealing with them well is important
- ▶ Building it right gives you offline support as a side effect
- ▶ Everything looks instantaneous
- ▶ Caching well will save you a lot of server side resources

**Making it look easy is
actually very hard**

Basic



Advanced



Split UI and Sync logic

Church and State

- ▶ Build your own local data store for in and outbound traffic
- ▶ Write or read to your local store first, then let your sync engine handle everything else in the background
- ▶ SQLite and Core Data can be complicated, but they'll save your users massive amounts of time, use them!
- ▶ Edge cases can be intense, but that's your problem now
- ▶ Loading spinners are the devil, this will get rid of them

Watch out

there are a few caveats

- ▶ Making things look instant means lying to your users (slightly) do so with extreme care
- ▶ Not all user data is equal. Prioritize
- ▶ Watch out for race conditions
- ▶ Beware the endless retry
- ▶ Audit your connections. Data is precious + expensive

The Data Myth

Kill all loading spinners

Help Maintain the Myth

Build your sync engine with care

**Treat your user's attention with
respect, and they'll give you
more of it**

Questions?

Simplicity

Simplicity

is making good choices

- ▶ Remove as many 'decisions' as you possibly can
- ▶ Make them choose between great things
- ▶ Reward their good behavior, encourage them to be awesome
- ▶ Cut down on cognitive load, use clear icons instead of text
- ▶ Be wary of over-simplifying, you'll confuse your users
- ▶ Consistency is actually more important than Simplicity especially when it comes to navigation

“Just add a Setting!”

Settings are a Copout

...most of the time

- ▶ Don't offload complexity on your users
- ▶ Lots of settings are often an indication of indecision
- ▶ Don't blame them for making bad choices, dealing with complex issues is your problem, not theirs
- ▶ Good settings are choices between great things or
- ▶ A chance to make your application their own
- ▶ Your app should always be functional by default

**“Users don’t know what they
want, I do!”**

Users don't know

...but neither do you

- ▶ Your gut is crap at building software
- ▶ Install Flurry, Localytics, Mixpanel, or Google Analytics
- ▶ Figure out how to track the behavior you want
- ▶ Watch out for short term number bias
- ▶ Invest the time for good, consistent event tracking
- ▶ Numbers are your only weapon against bad ideas, especially your own

Simplicity

Remove Decisions
Make Awesome Settings
Validate your Assumptions

**Simple, Beautiful
software takes time,
but the payoffs are
immense**

Questions?

Design

**Designers: Customize
Everything
Developers: Customize
Nothing**

Innovate on Purpose

Break the rules, learn them first

- ▶ 100% adherence to the standards defaults is boring
- ▶ 100% customization is confusing, particularly with navigation
- ▶ Designers, be aware of the costs of your customizations
- ▶ Developers, communicate time concerns
- ▶ Devs, never say "No". Designers, never accept "No".
- ▶ For Reference: One Android developer at Tumblr spends one week per release on margins alone. We're working on fixing that.

**The designs should be
final before developers
build them**

Designs are never final

Good design requires dialog

- ▶ It's not finished until it ships
- ▶ Remember, change requires TIME!
- ▶ Developers, it's on you to call out major changes
- ▶ Pixel perfection is essential, but you should try to polish once
- ▶ Consistent, correct margins make the difference between good applications and great ones

Design

Design is a Conversation
Innovate with Purpose
Make Beautiful Things

Questions?

The Market

**Nothing beyond this point
is helpful unless you...**

Do

not

build

shitty

applications

...seriously

Talk to your users!

the app/play store is the only tool you have

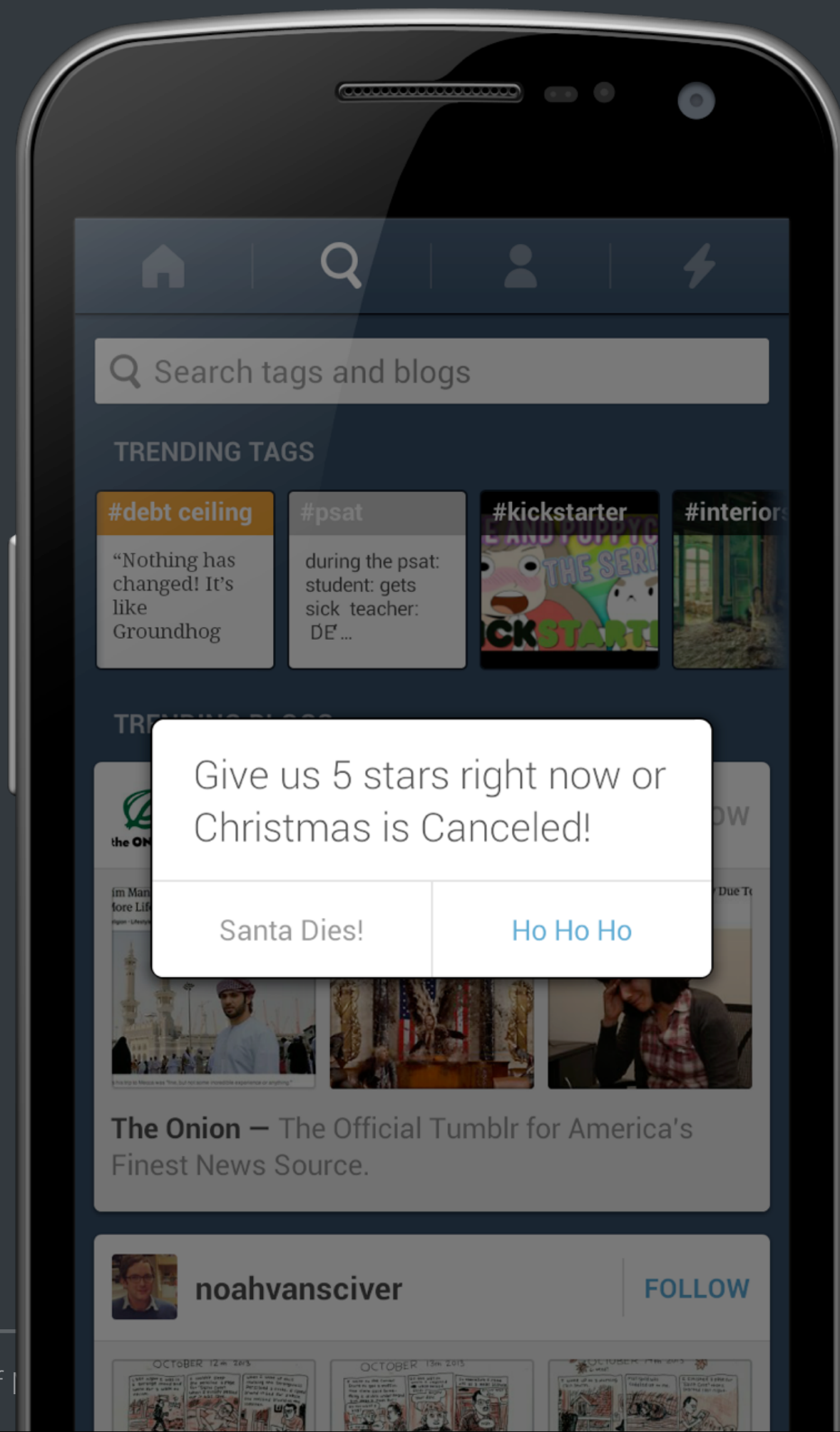
- ▶ A good rating requires building a community
- ▶ respond to reviews in the Play Store
- ▶ learn everything you can from the 1-star reviews
- ▶ Remember those analytics? Use them to triage the rage
- ▶ at tumblr... the feels run strong

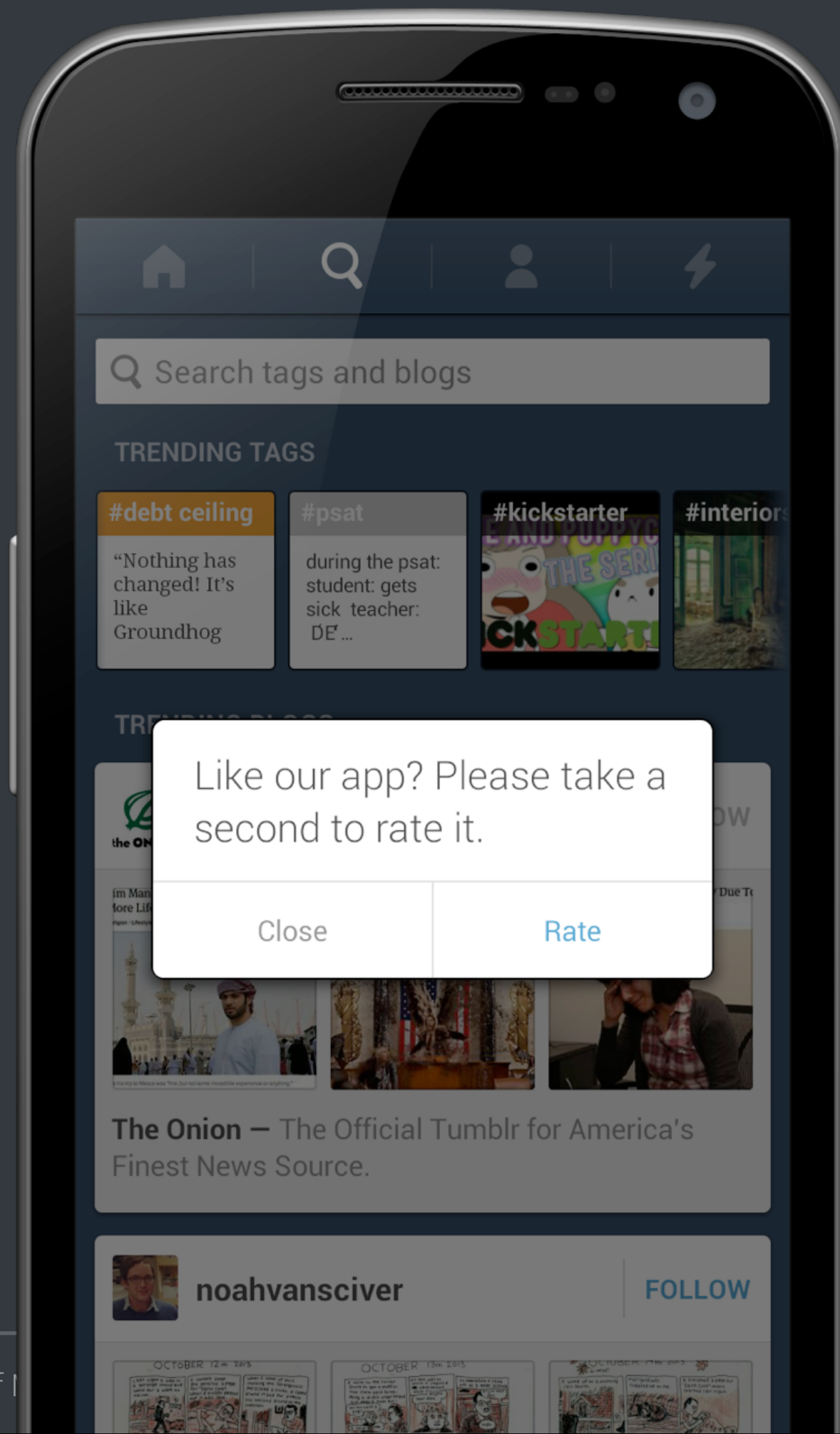
Rating Nags are Annoying

they are also effective

...incredibly effective

- ▶ Our daily average rating went from 3.6 to 4.3!
- ▶ Timing is *everything* catch people at a happy moment
- ▶ Be glib, positive, and friendly in your language
- ▶ *Never* ask them more than once in a 6 month period
- ▶ Ask for help, don't demand it, but don't beg for it either
- ▶ Your users are rooting for you, help them express it!





Like our app? Please take a second to rate it.

Close

Rate

The Onion — The Official Tumblr for America's
Finest News Source.



noahvansciver

FOLLOW



The Market

Engage your Users
Learn from your Mistakes
Ask your users to show
some love

Questions?

**We are all on the verge of
something great**

Respect your users time
Build Beautiful Software
Innovate with Purpose
Build a Community

**Make incredible things,
none of you has any
excuse to do otherwise**

Thank you!

haseman@tumblr.com
haseman.tumblr.com
[@haseman](https://www.tumblr.com/haseman)