

ROAD TO A GREAT PRODUCT



Methods, Tools and Hacks

@mxbraud

you're in charge of a
product

you have
users

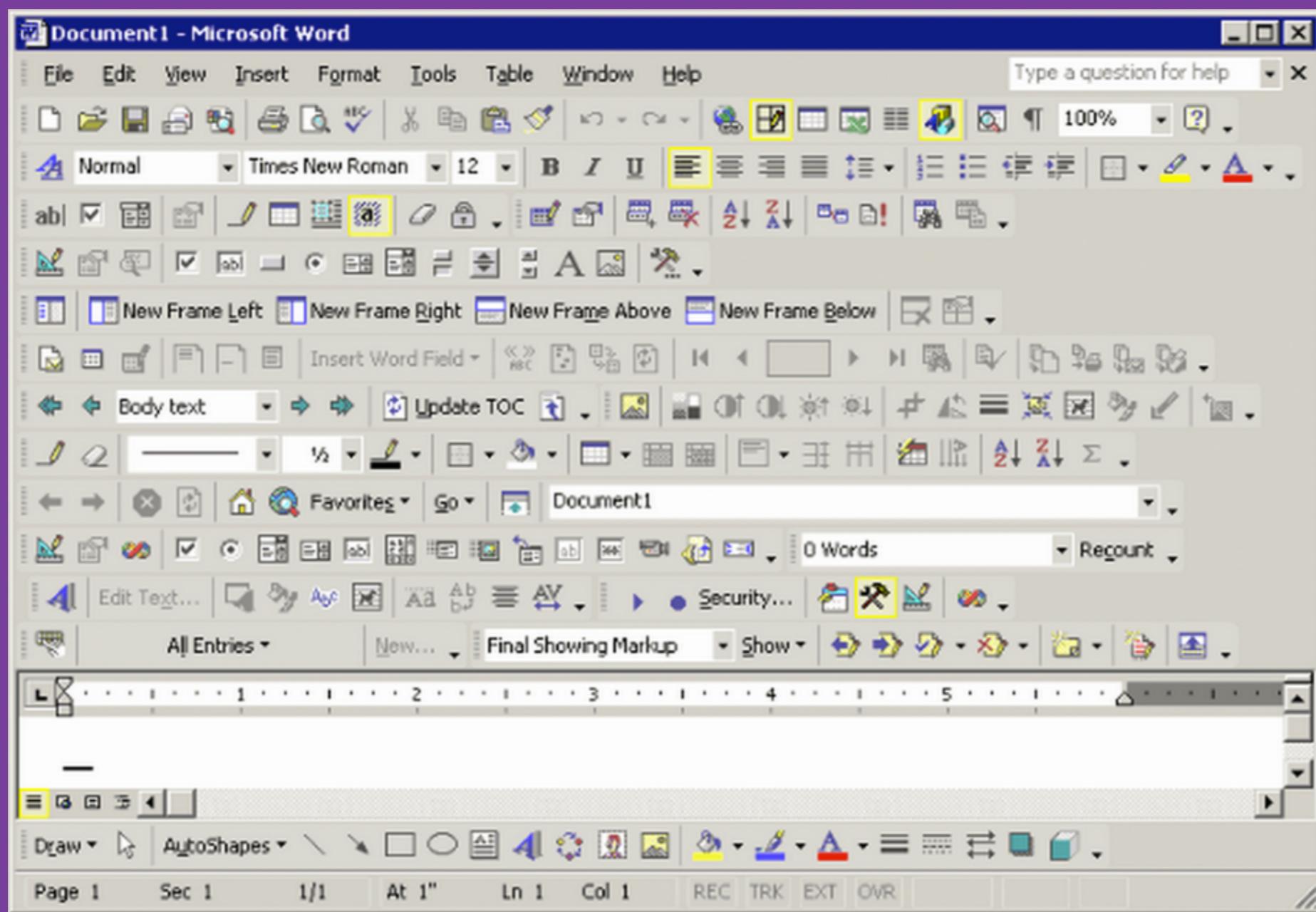
you're ready for the
next level



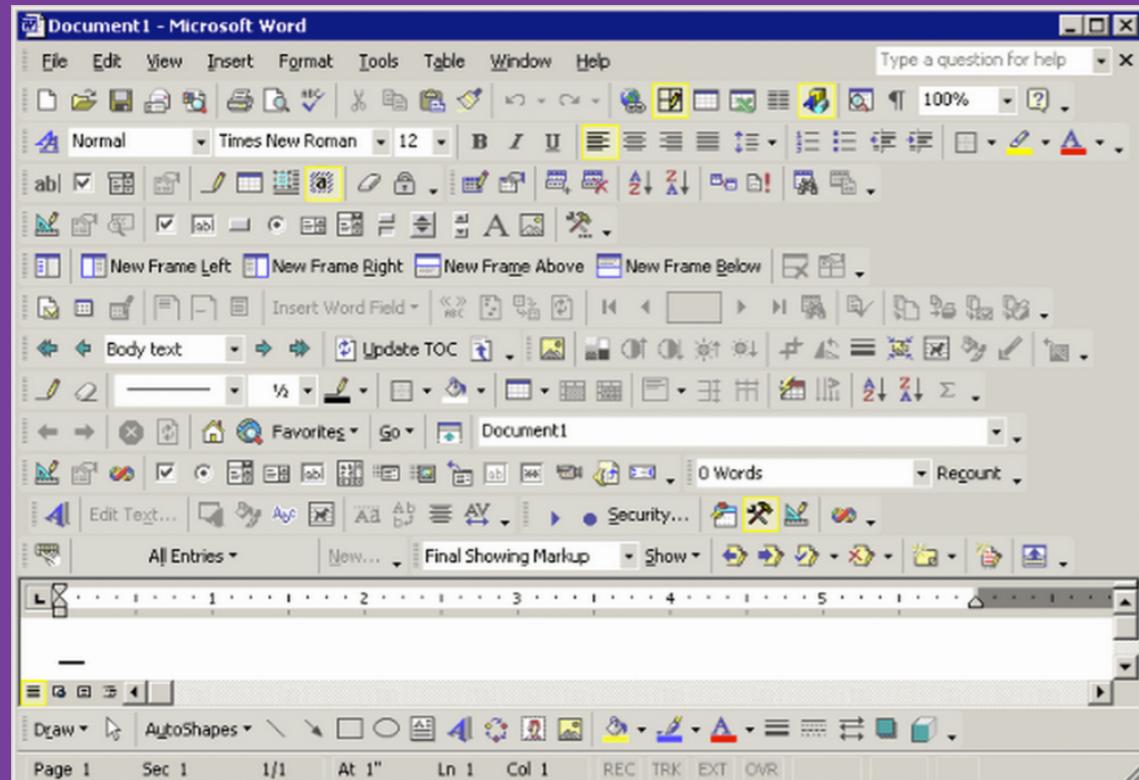
“If you want to be a
long-term success,
build a great product”

Sam Altman, Y Combinator

What great is not

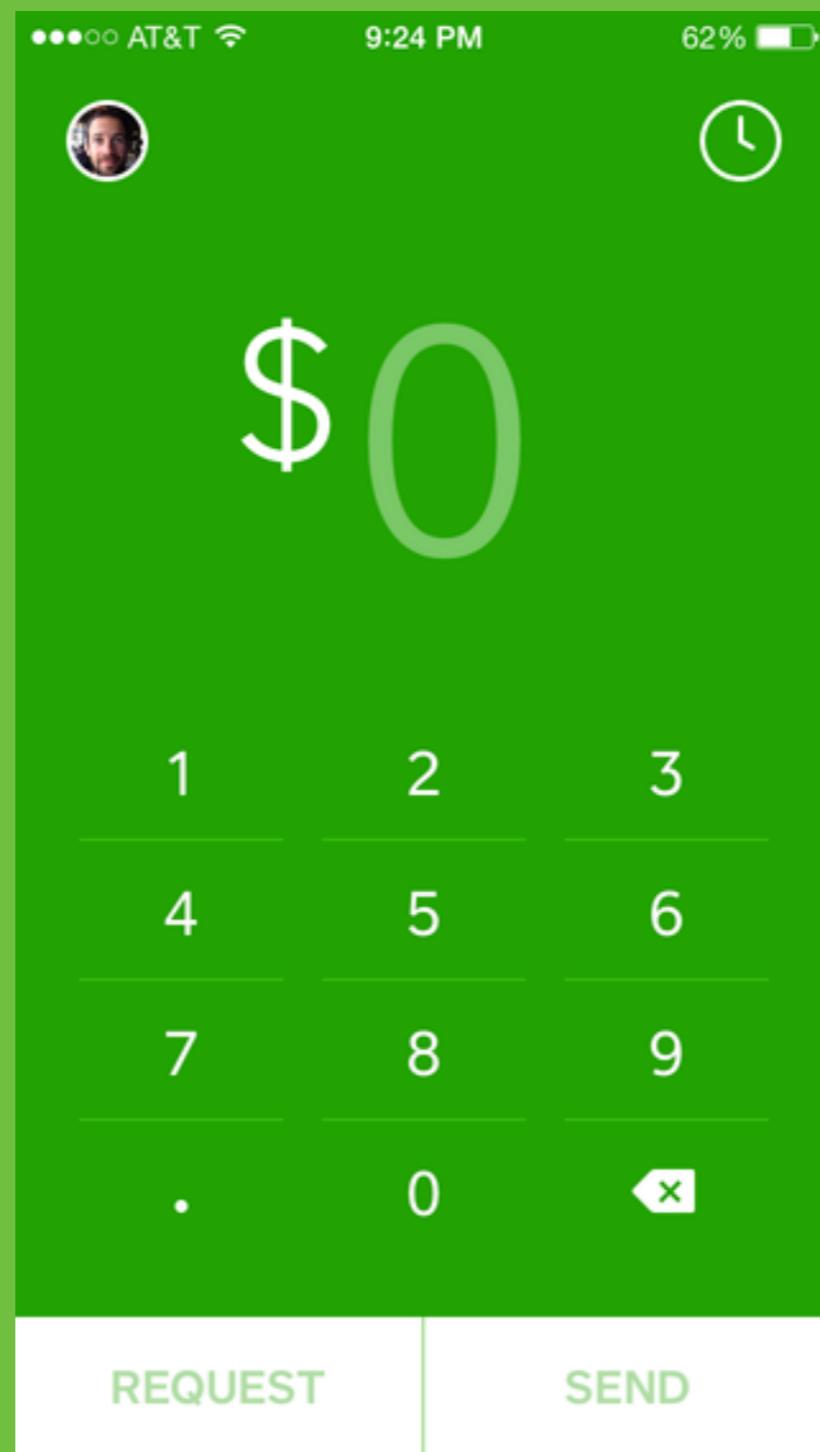


What great is not

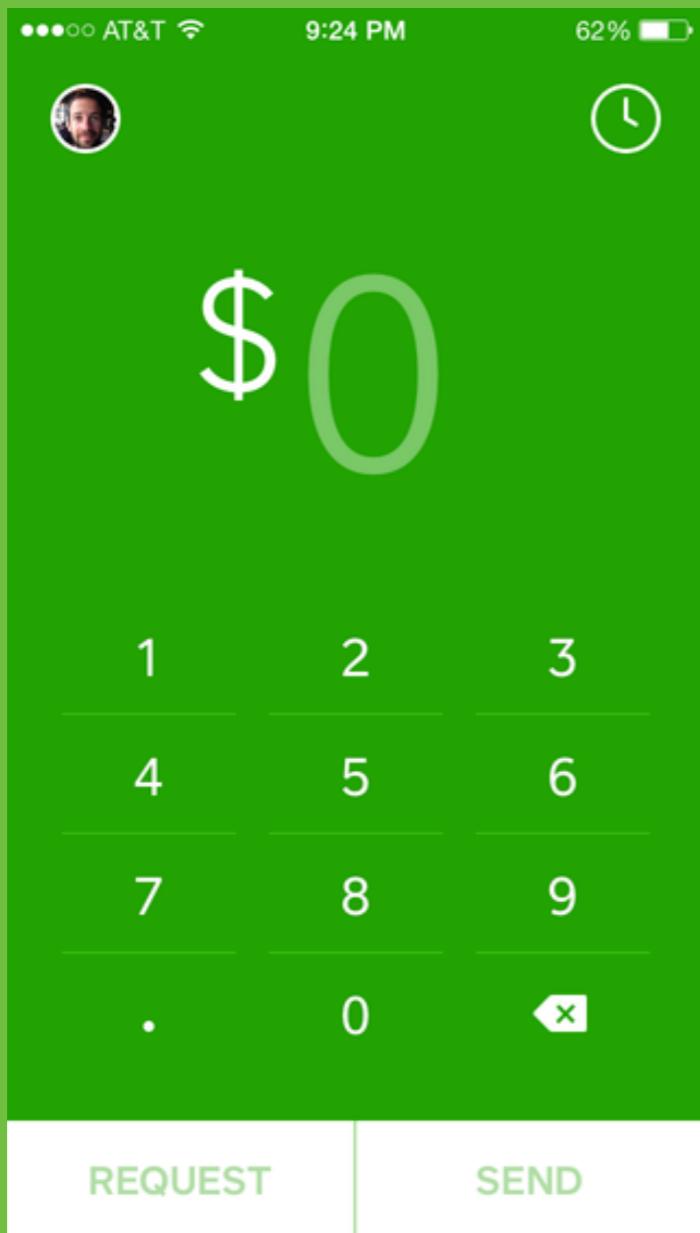


- more features
- more settings
- more uses cases covered

What great is



What great is



- less features
- easier to use
- more value
for 80% of use cases

How?

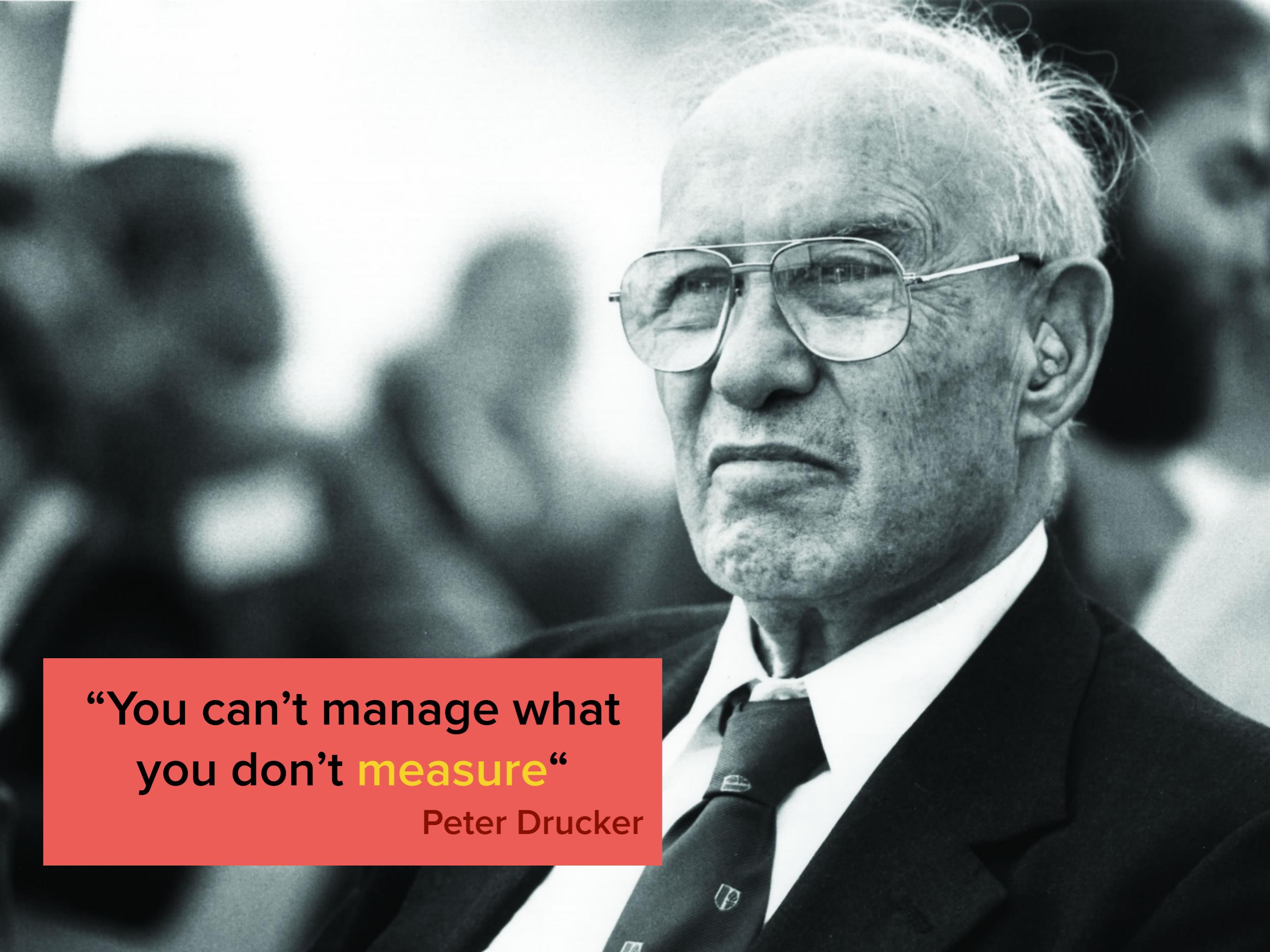
1. Understand your users
2. Select a feature
3. Design loops!
4. UI and code
5. Roll-out and analyse

it's not difficult :)

just follow the steps!



Understand your users

A black and white photograph of Peter Drucker, an elderly man with white hair and glasses, wearing a suit and tie. He is looking slightly to the left. The background is dark and out of focus.

**“You can’t manage what
you don’t measure”**

Peter Drucker

We are going to measure 3 things

NPS

App Ratings

Retention



if your product is
an app!

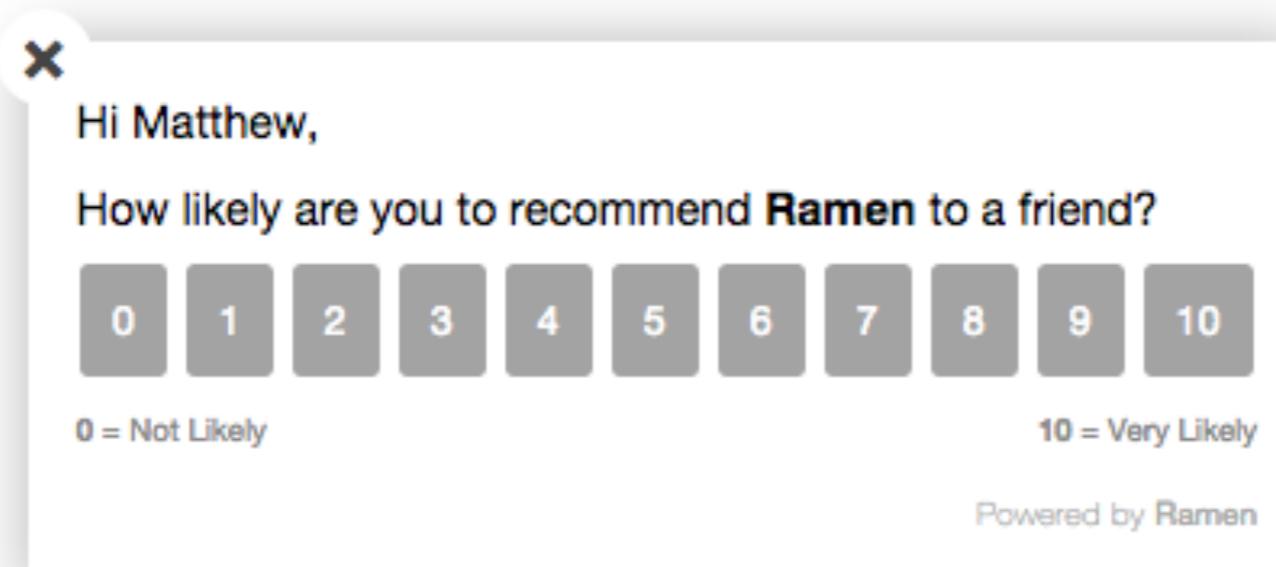
NPS - App Ratings - Retention

How likely are your users to
recommend your product to a friend ?

NPS = Net Promoter Score

NPS - App Ratings - Retention

Implement it in your product
from scratch or use **RAMEN**



NPS - App Ratings - Retention

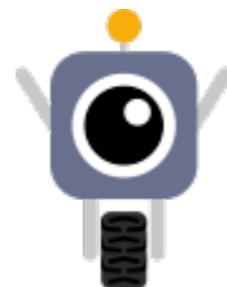
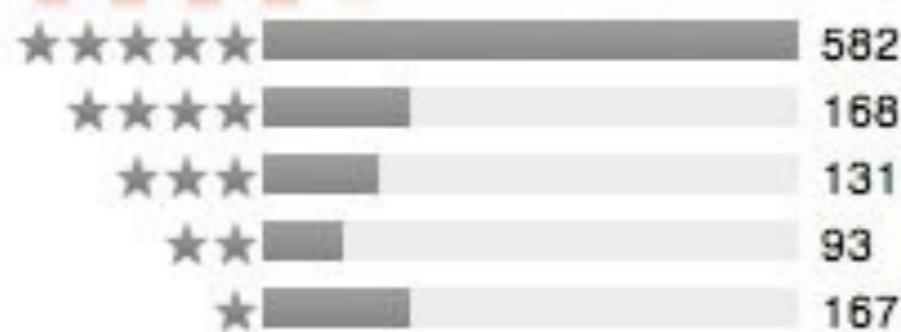


check the ratings
for each release

Customer Ratings

▼ Average rating for the current version:

★★★★★ 1141 Ratings



use
appbot
to automate it!

NPS - App Ratings - Retention

How many of your users are still
here **after 1 week?**

After one month?
After three months?

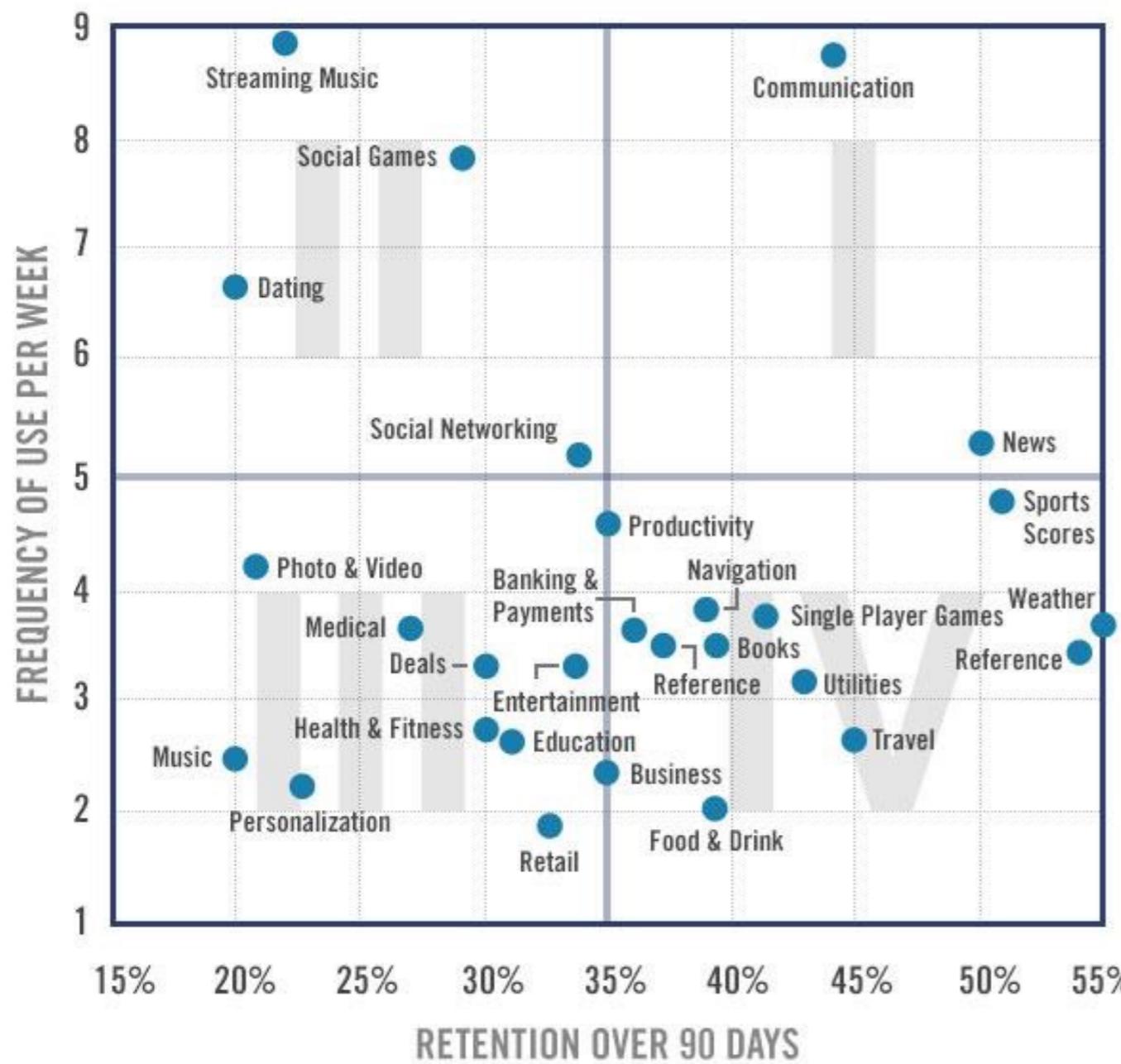
NPS - App Ratings - Retention

Use [AMPLITUDE](#) or [mixpanel](#)
to measure it

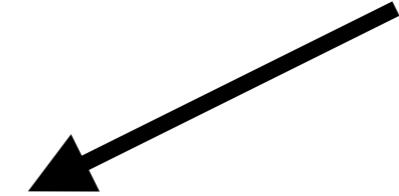
Start Day	Users	DAY 0	DAY 1	DAY 2	DAY 3	DAY4
Jul 22	12.233	100%				
Jul 21	13.552	100%	57.6%			
Jul 22	9.341	100%	60.4%	40.9%		
Jul 19	6.419	100%	43.0%	42.8%	31.1%	
Jul 18	7.435	100%	28.0%	28.6%	29.9%	22.4%
Jul 17	12.467	100%	32.4%	19.2%	21.3%	23.3%
Jul 16	11.535	100%	57.0%	23.6%	14.7%	18.0%

NPS - App Ratings - Retention

Loyalty by Application Category

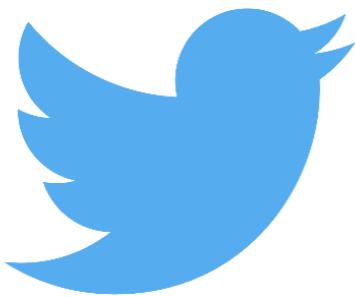


This is what you
should expect
after 3 months



NPS - App Ratings - Retention

Some social media examples:



31%



33%



48%



77%



98%

Retention at 90 days,

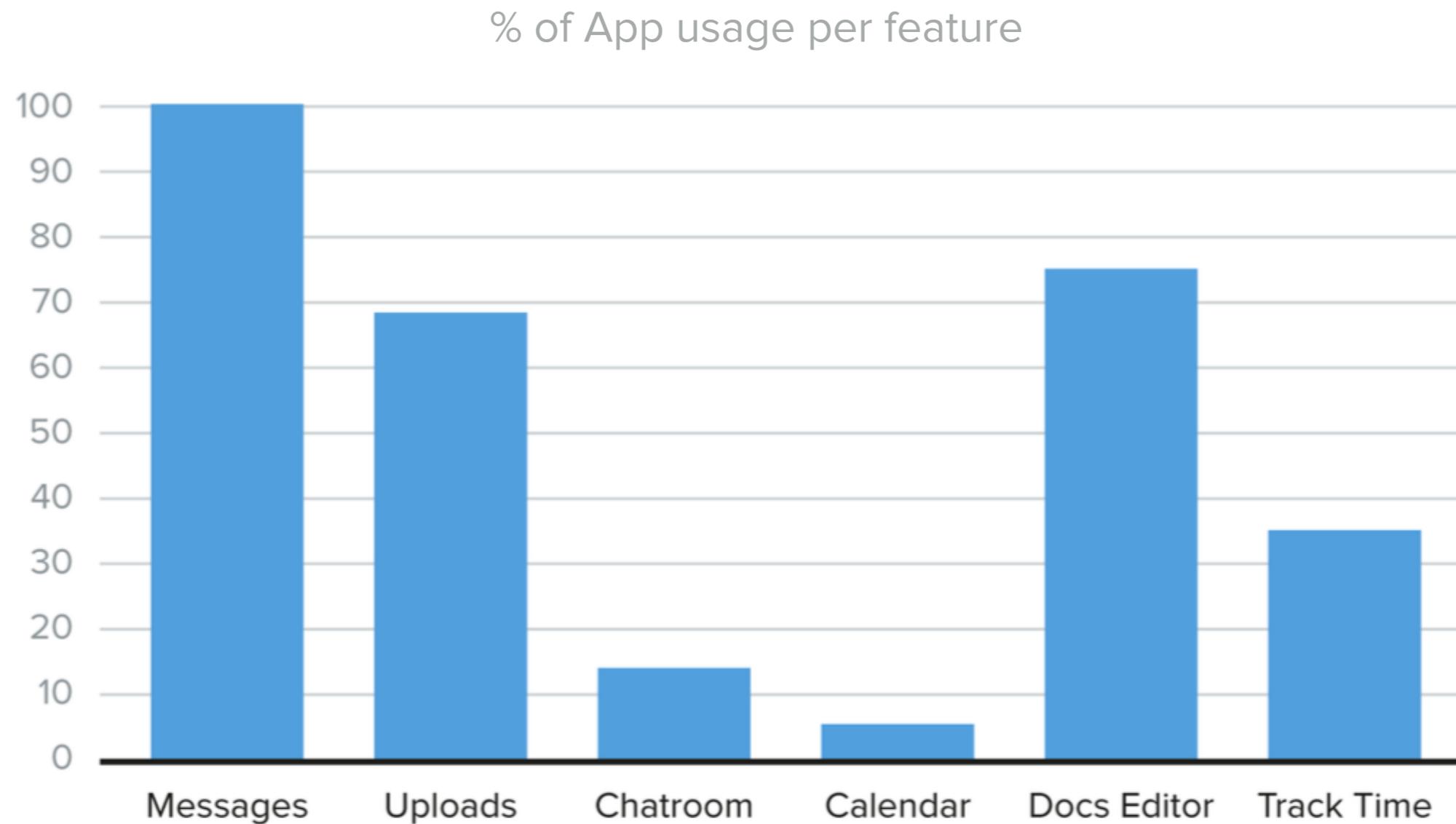
insane!





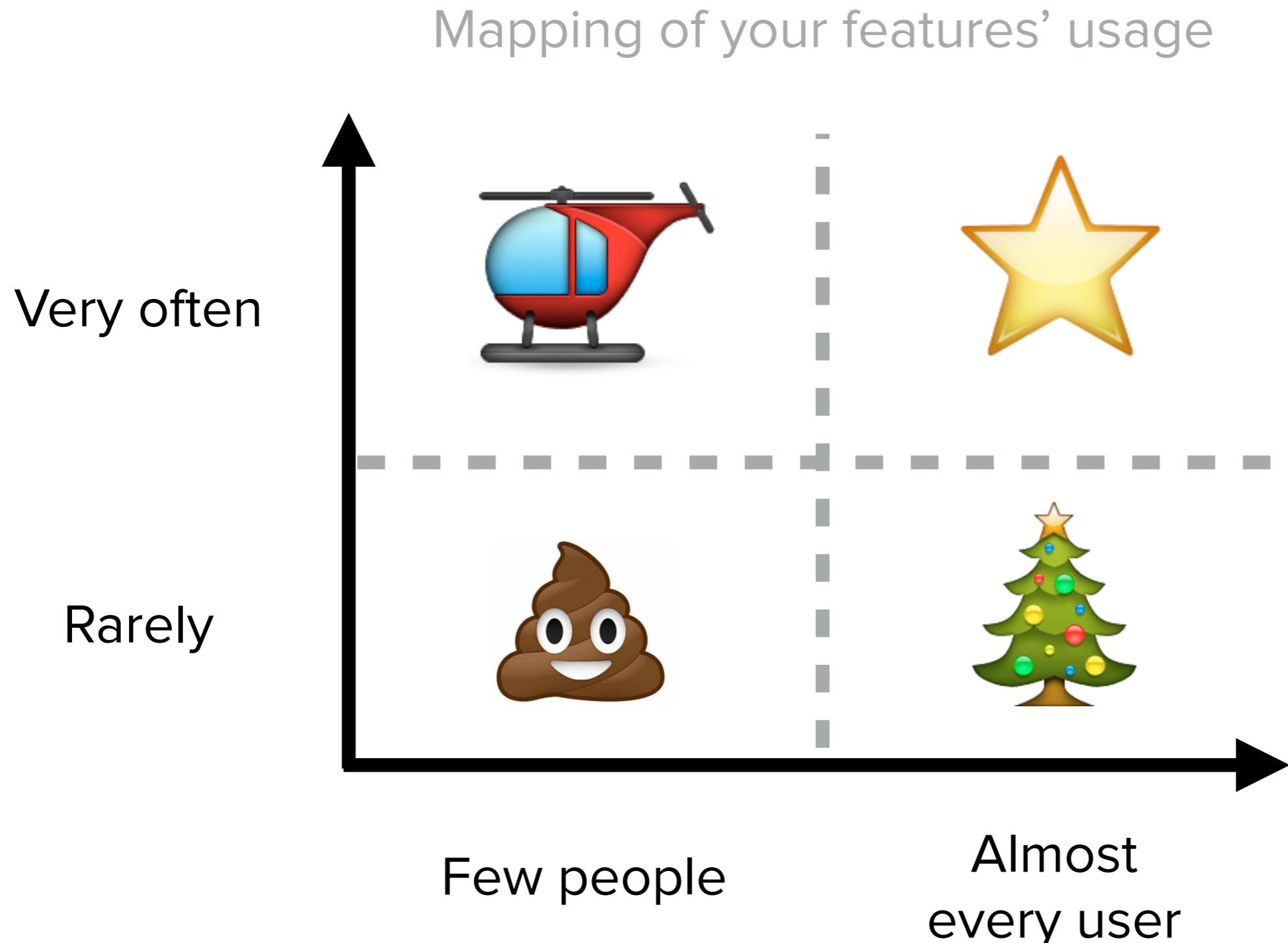
Select a feature

Analyse your features' usage



Do it with Amplitude or Mixpanel too!

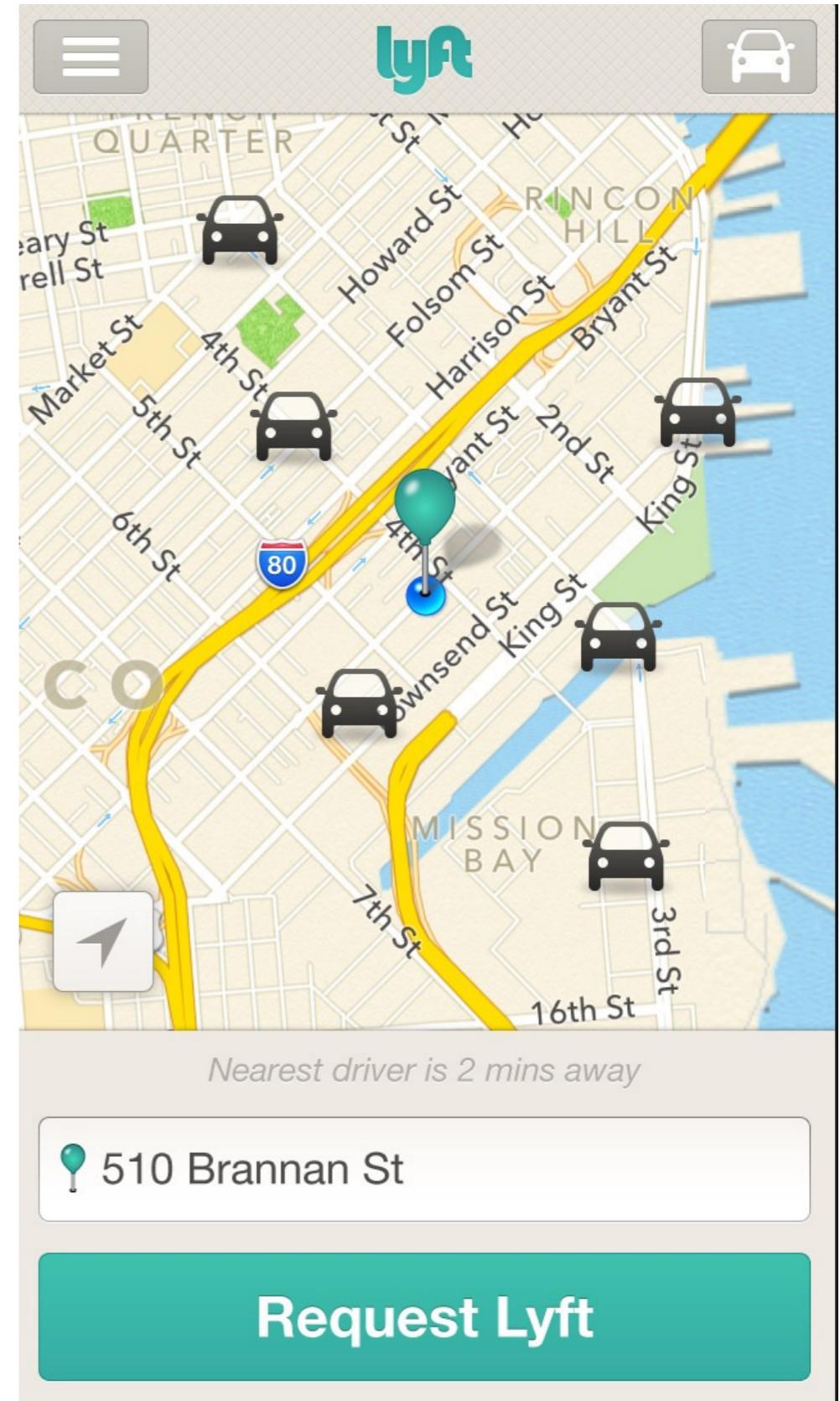
Map your features into 4 categories





Key feature of your product

example:
requesting a ride in Lyft





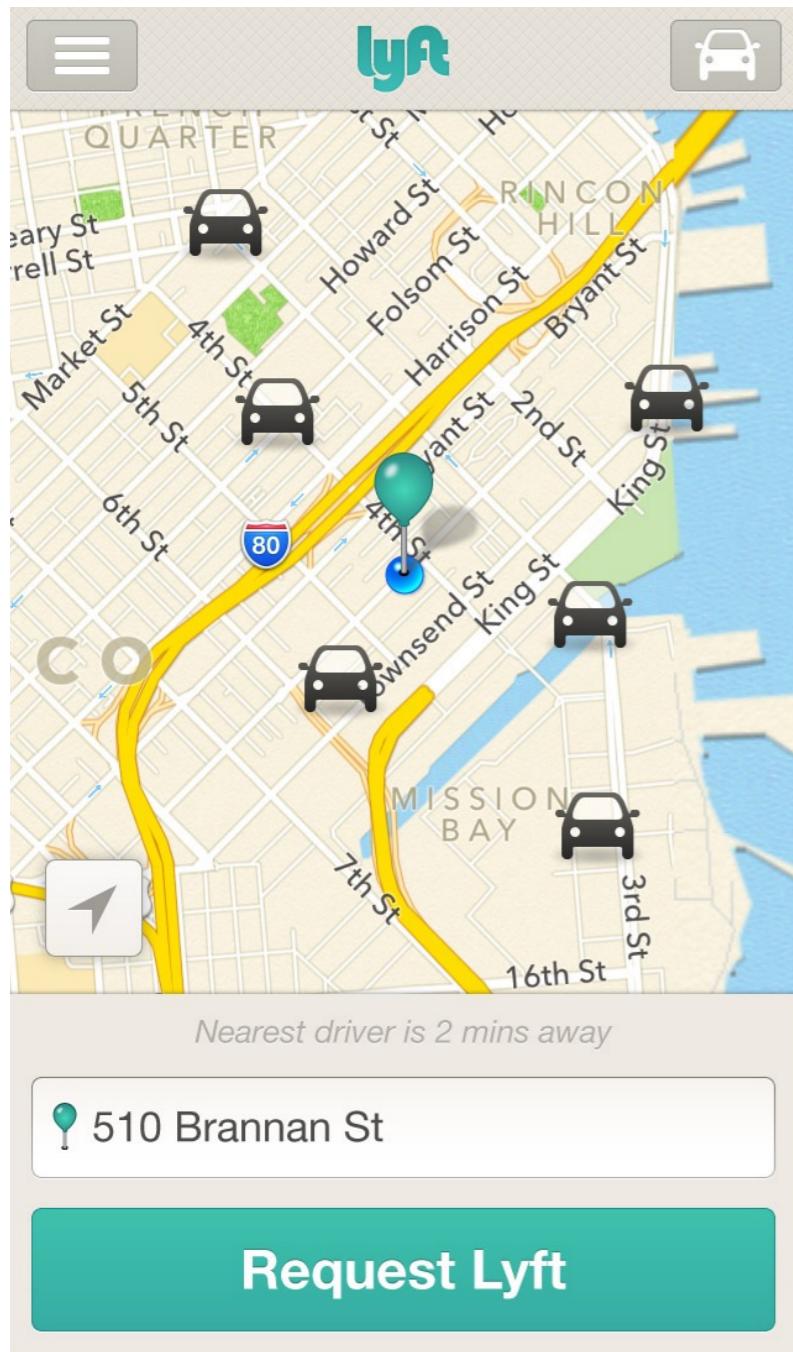
Always make it better.

- easier to use
- delivers more value
- nicer design

A slight positive change is a huge plus for your product.



Lyft tested designs every week to improve their main feature: requesting a ride.

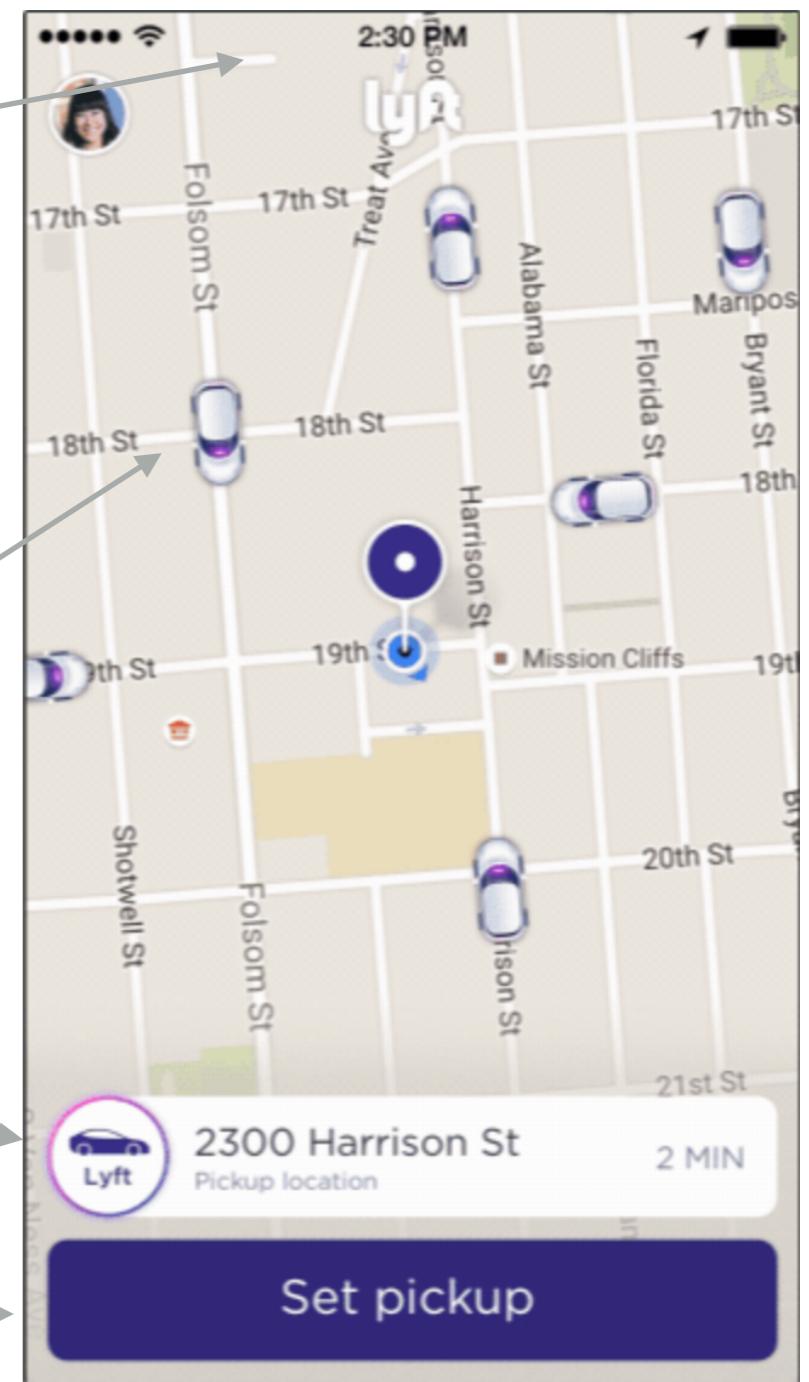


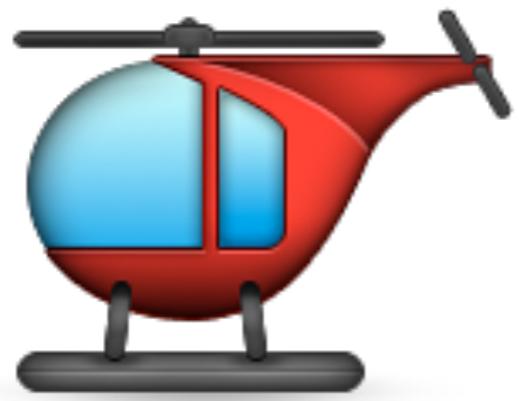
less options to
show more map

more accurate cars

car and pickup info
to reassure user

new CTA text and
color for higher
conversion





Feature a small
part of users
use a lot

Maxime Braud
Just now

selling my helicopter
€500 — Stanford, California



Mark as Sold

Like Comment

example:
selling and buying stuff on Facebook



Is there a **discoverability** problem?



Make sure your users
know this feature exists



“When Microsoft asked their users what they wanted added to Office, they found 90% of the requested features were already there.”

Intercom on PM



Facebook is A/B testing Market offers as a Tap Bar option on mobile

••••• Meditel ⌂ 09:35 ⚡ 100 % 🔋

Find new things to buy

Discover, explore, buy

Discover all the new things from your sales groups and Fan Page shops

OK

More offers notifications >

YOUR SALES GROUPS

Newsfeed Friends Market Notifications Plus

3 likes

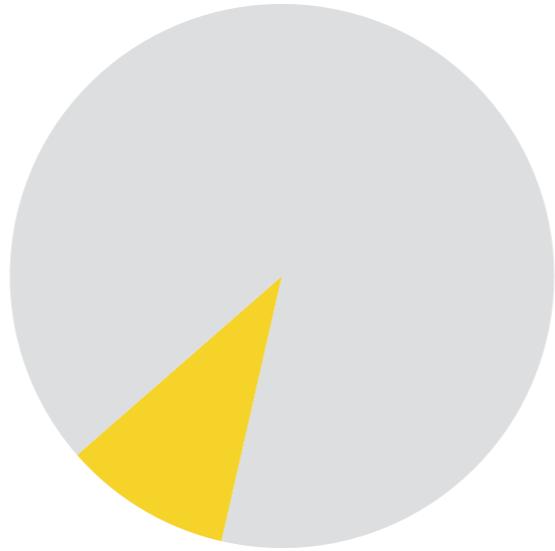
Market 1

Notifications

Plus

YOUR SALES GROUPS

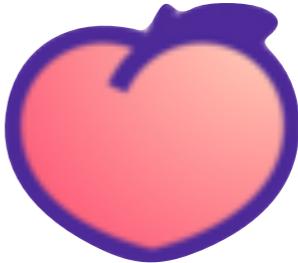
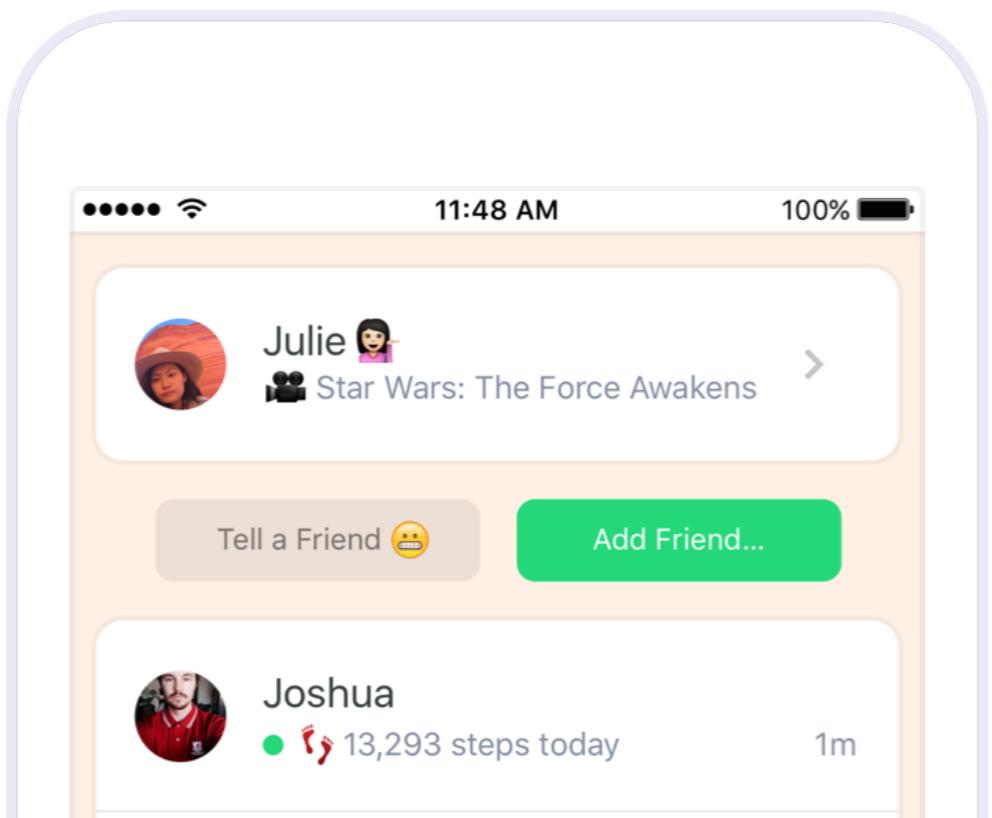
Newsfeed Friends Market Notifications Plus



If users all know it but still
don't use it, maybe you're
building something for a
too small part of your target



Dangerous because someone can make a **more focused product**, better at your core function, and take your users.



peach is an attempt of a more focused FB



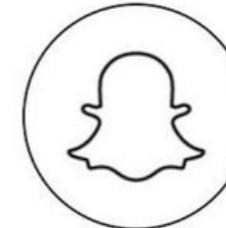
A lot of users use
it, but not often

example:
Snapchat Discover

••••• VERIZON 10:20 7 12%



discover





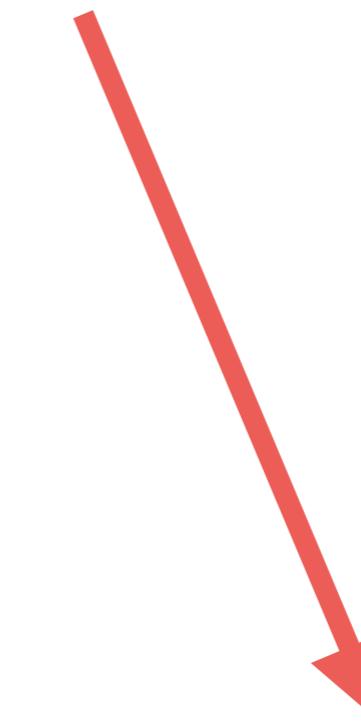
Will you benefit from users using it more?

yes



no

ex: settings



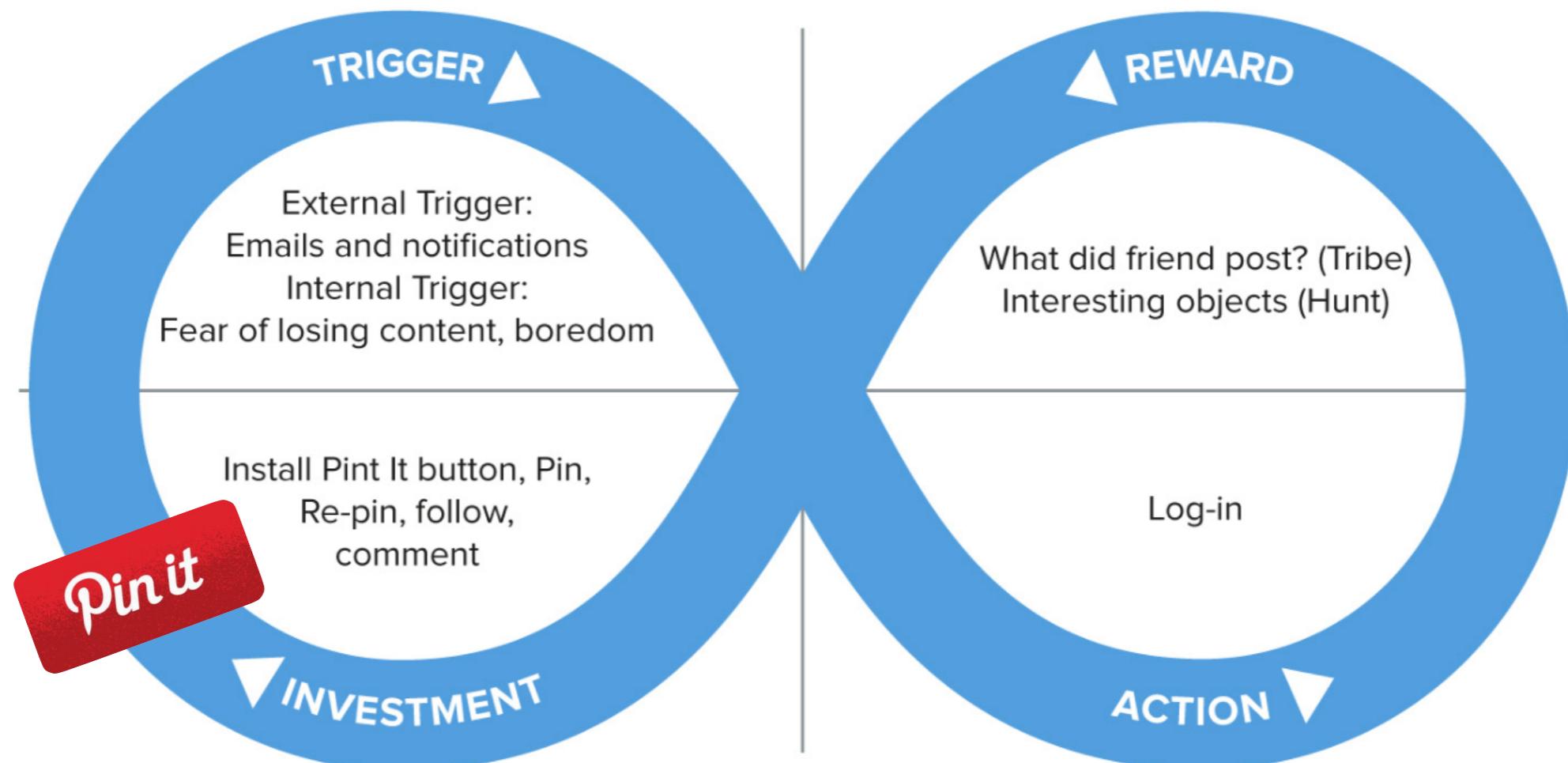
work on your hook

see next slide :)

do nothing



Here is the hook **Pinterest** used
to make users come back more often





focus on **investment**

make users come back to
your feature!

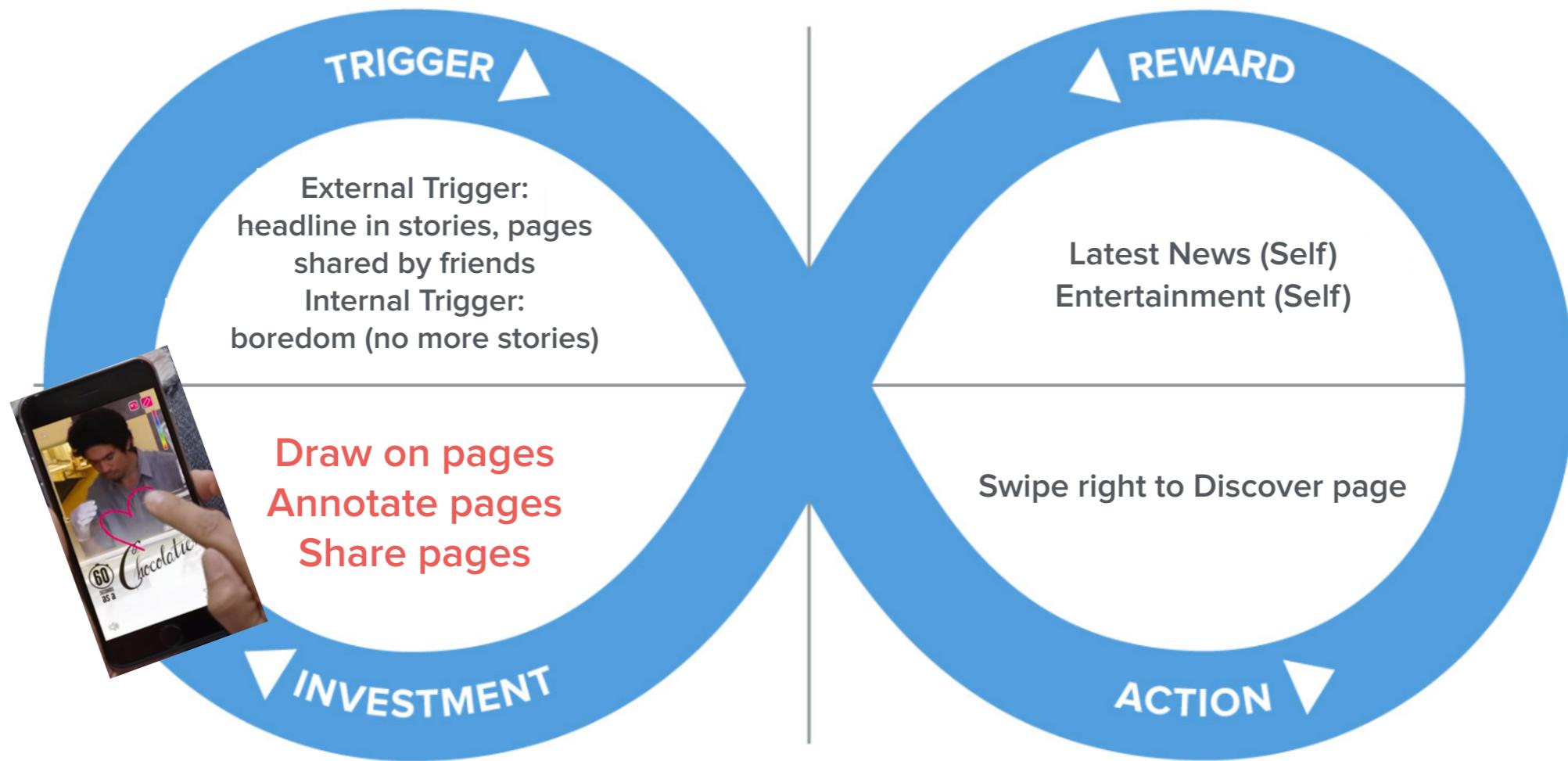


**Snapchat now lets users
share and draw on
Discover stories**





Snapchat used this hook to boost Discover usage which is their main way to generate revenue





Feature nobody uses, ever.

Kill it. Now.

Don't spend more time and money building
something nobody wants!



example: Evernote Chat

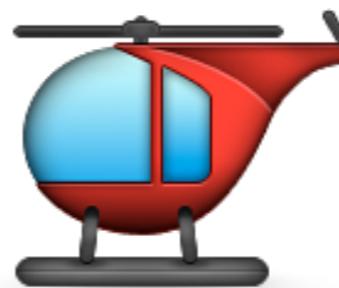
The screenshot shows the Evernote Basic interface. At the top, there's a header bar with the text "Evernote Basic". Below it, a search bar contains the placeholder "New note in Maxime Braud's notebook". To the right of the search bar is a button labeled "New chat" with a speech bubble icon, which is highlighted with a yellow box. Further to the right are "Search in..." and "More" buttons. Below the search bar is a toolbar with several buttons: "+ New notebook", "Find", "Owner" (which is currently selected), "Name", "Count", "Date", and a grid/icon button.

Do we really need to chat in Evernote?
Interface would be better with less buttons

Select one feature for the next step



or



or



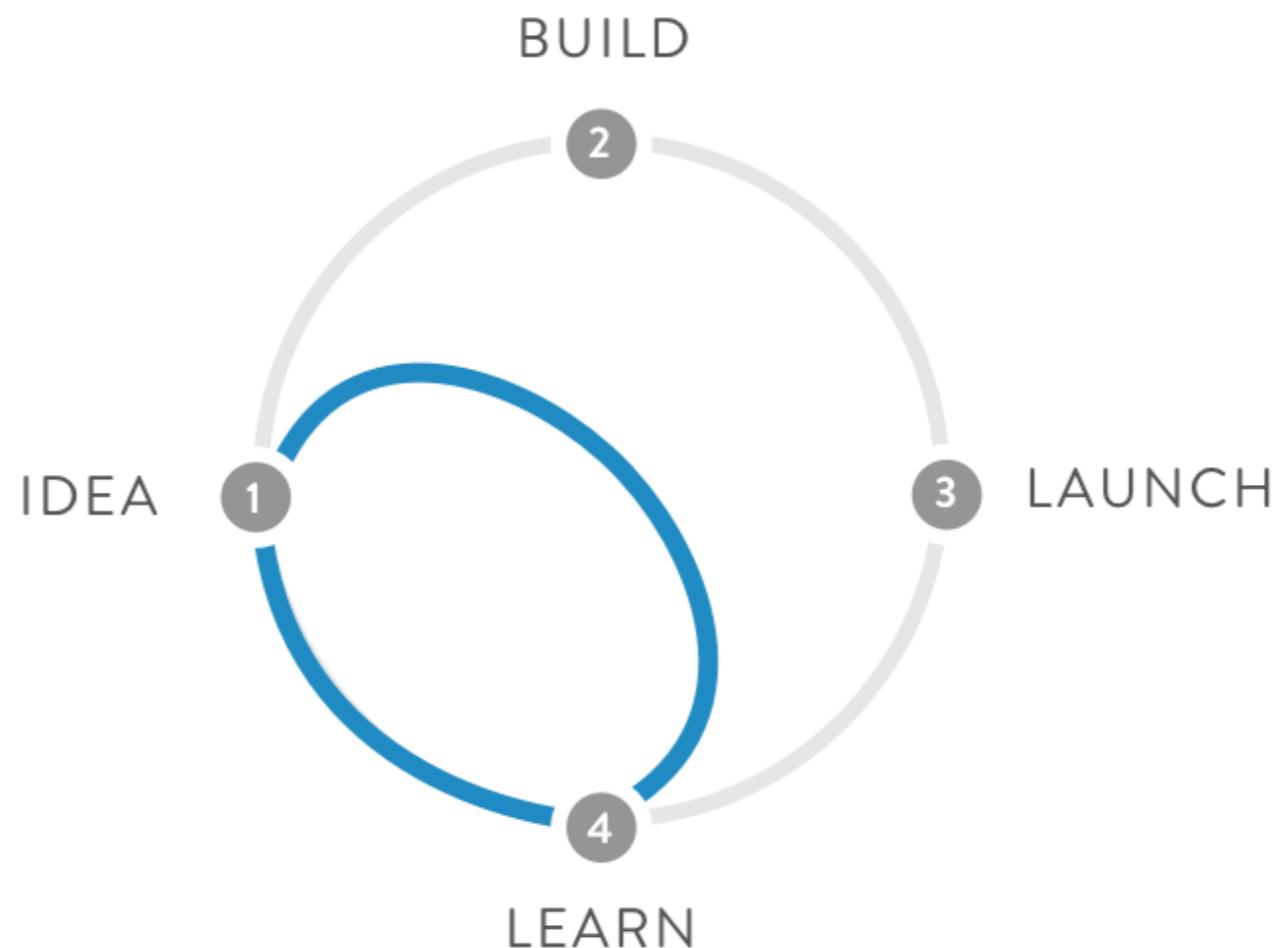
only one!



Design loops

1 feature only
1 week per loop

The loops make you iterate fast
to find the right feature,
before building anything



4 steps

dig - draw - prototype - test



dig - draw - prototype - test

Gather all the
information you can on
the user perception
of the feature

dig - draw - prototype - test

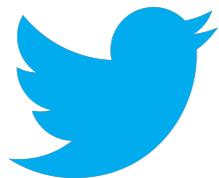
Dig into:



support@ emails



app store reviews



tweets

dig - draw - prototype - test

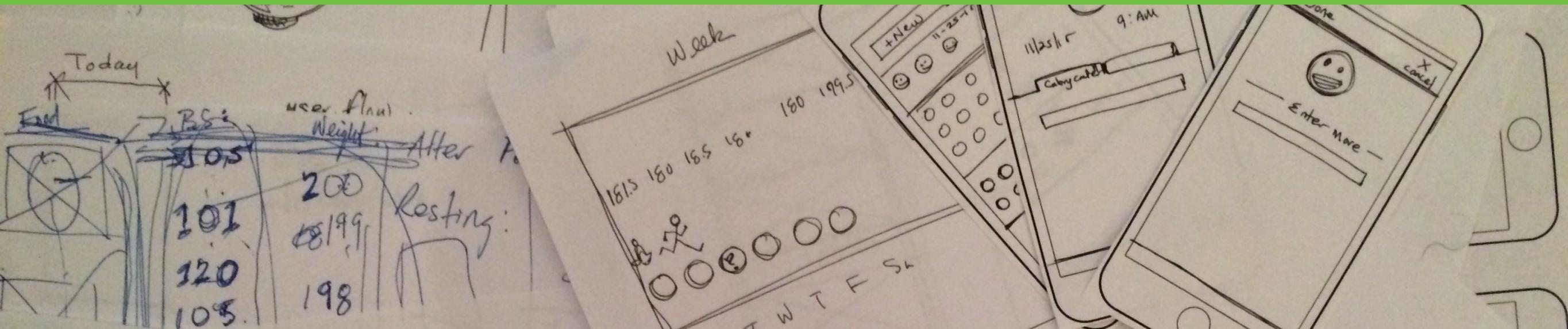


**Call or get a coffee
with a few users**

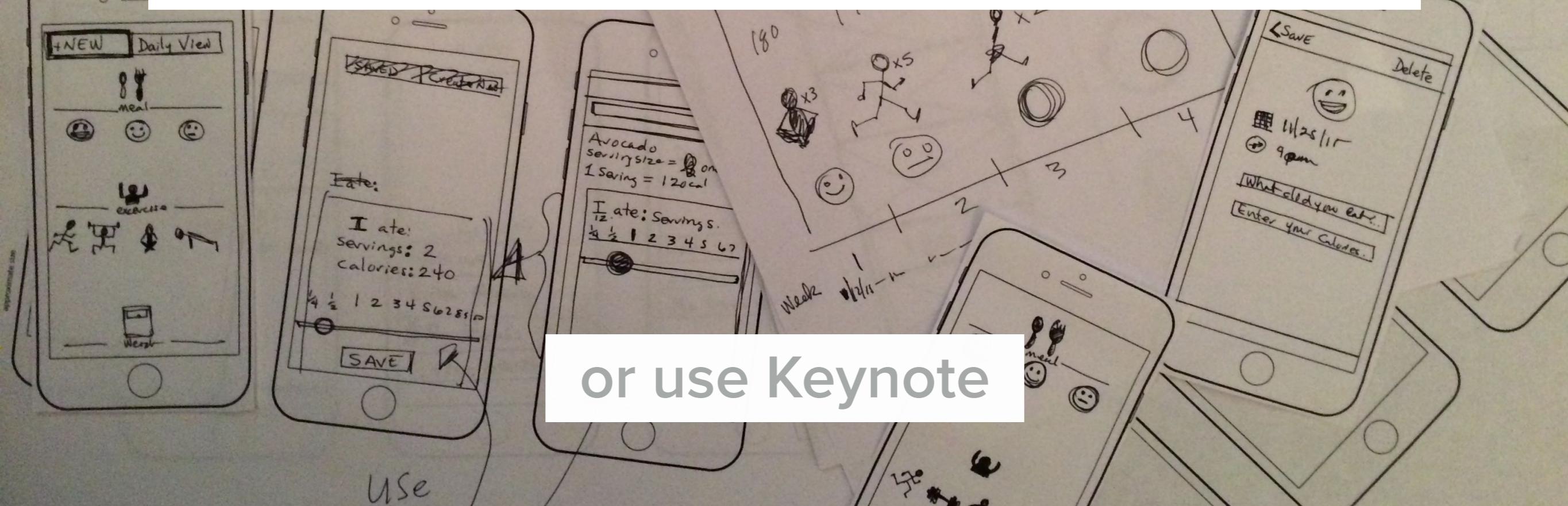
dig - draw - prototype - test

**Listen to their problems,
not their solutions!**

dig - draw - prototype - test

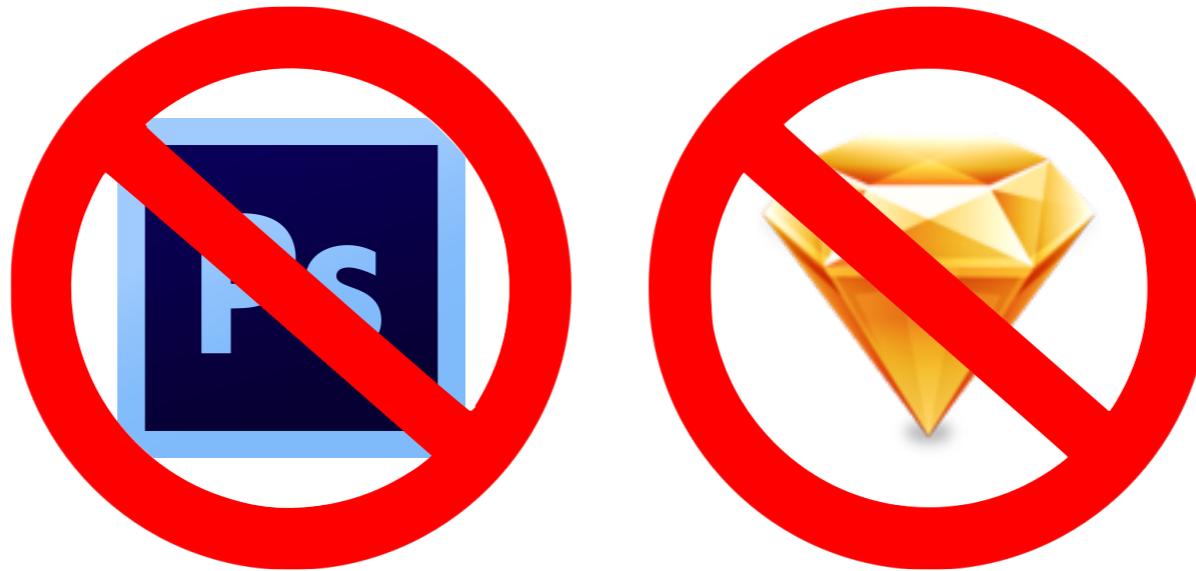


Draw your solution on paper



or use Keynote

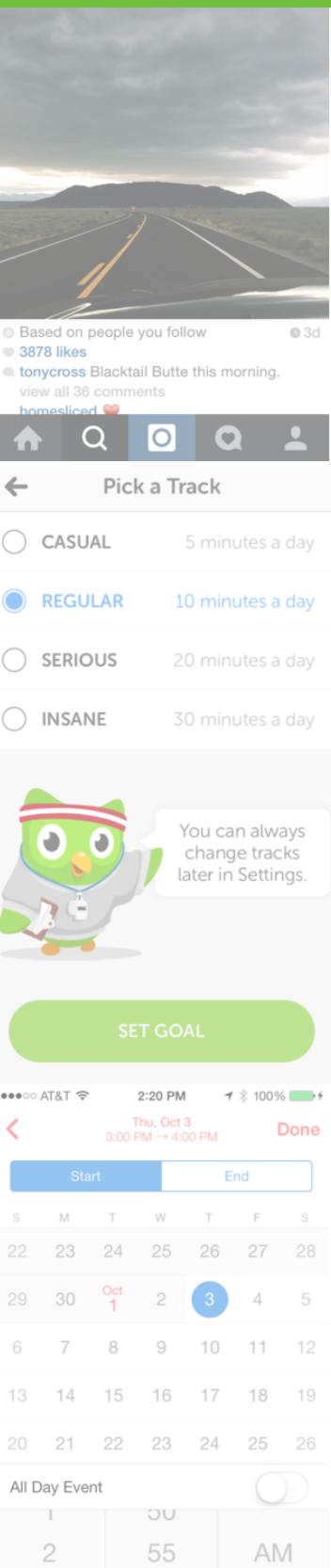
dig - draw - prototype - test



No Photoshop or Sketch at this point!

Focus on content, flows and
interactions, not on colors

dig - draw - prototype - test

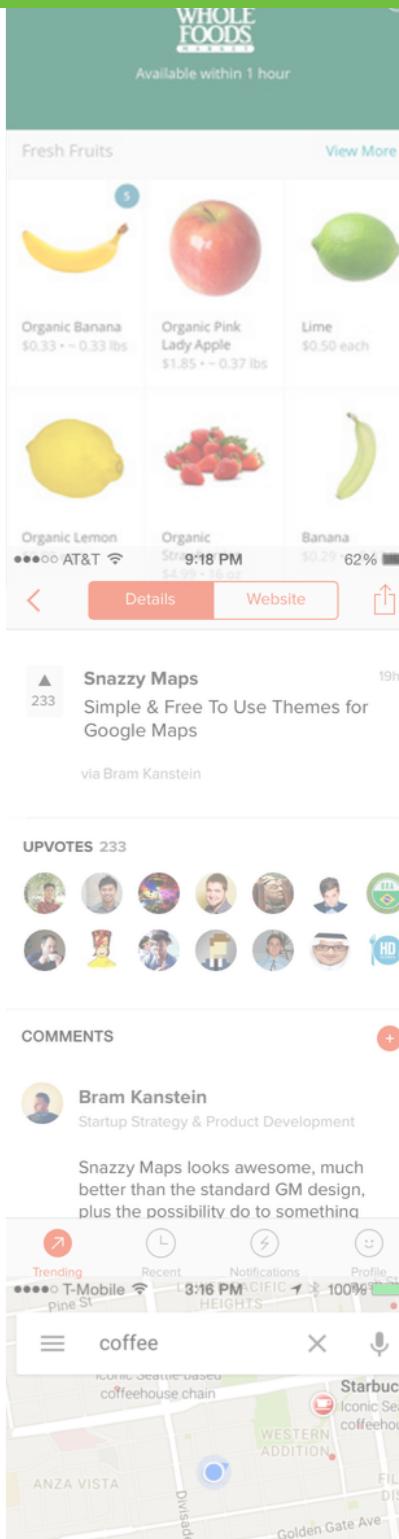


Get inspired
by the best flows

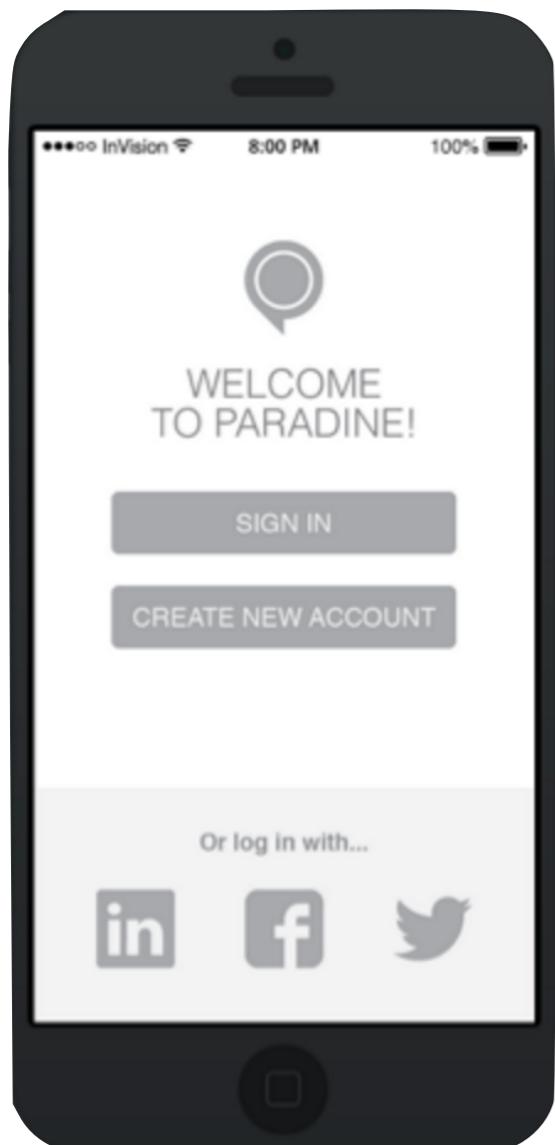
visit

uxarchive.com

littlebigdetails.com



dig - draw - prototype - test



Create a prototype of your feature

A prototype is something users can **experiment**, not just look at

dig - draw - prototype - test

**Design directly in the browser
by using front-end frameworks
such as**



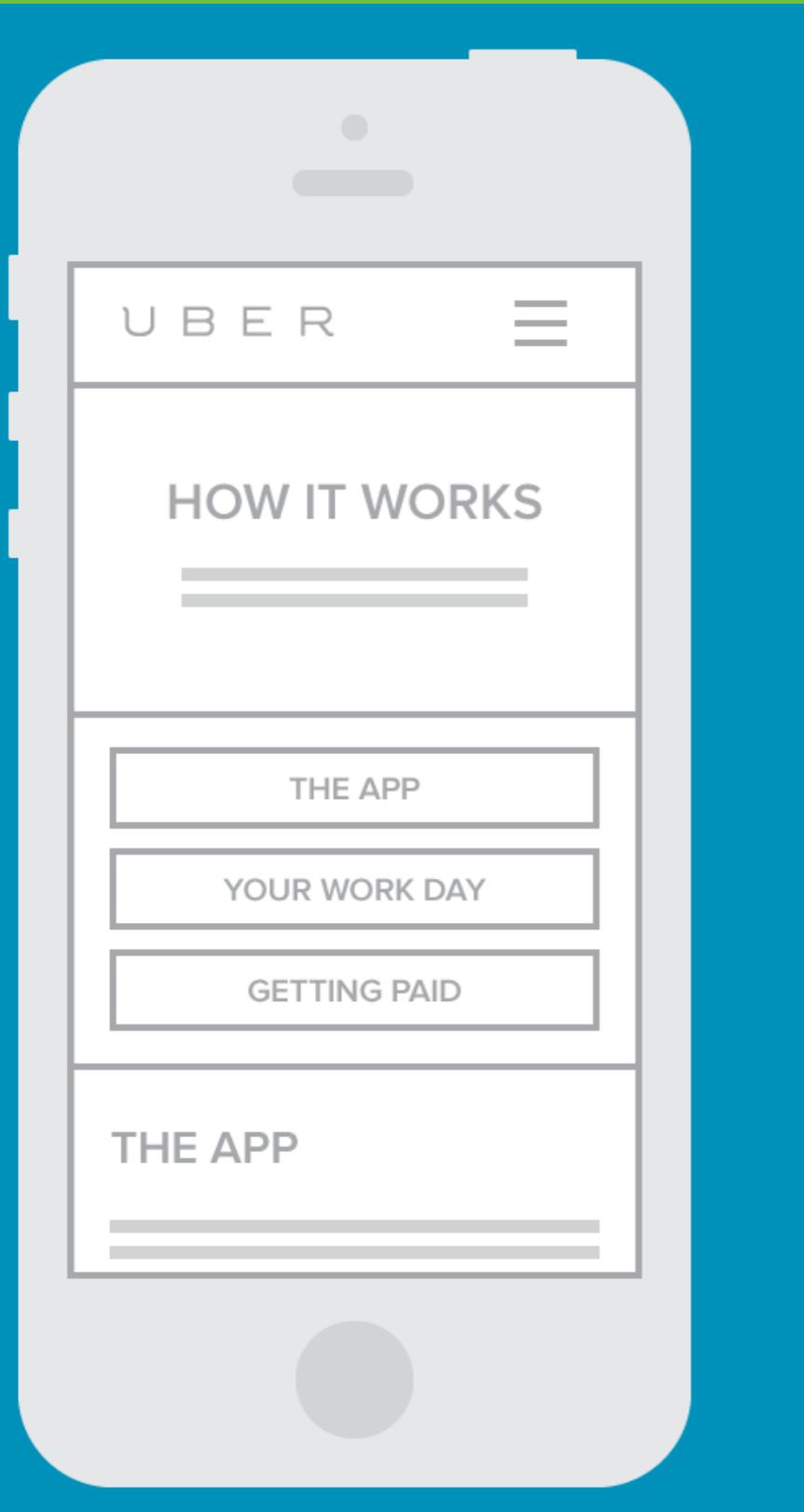
Bootstrap



Foundation
Start here, build everywhere.

this requires some basic coding skills

dig - draw - prototype - test



Designing in the browser
is awesome because you:

- **test a realistic experience w/ interactions**
- **use real content**
- **iterate fast using SCSS variables**
- **can test on different devices (mobile, tablet, desktop) and browsers**
- **can re-use your code in production**

dig - draw - prototype - test

or use a prototyping tool like



PrincipleForMac.com

convenient for animation-heavy apps or
if you can't code

dig - draw - prototype - test

**1) Test first among your
coworkers and friends**

dig - draw - prototype - test

2) Organize a user test session

dig - draw - prototype - test



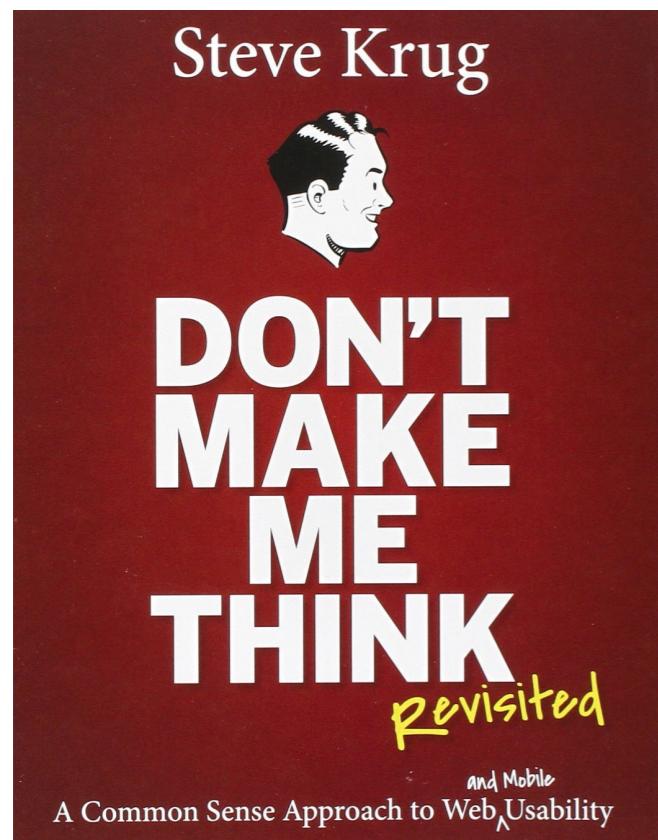
Invite 4 people

- Ideally targeted users, still ok if they're not
- Find them among your extended network or on public FB groups
- Offer them some free rides (if you're Uber), or a gift card

dig - draw - prototype - test

Typical User Test Session (45min)

One user at a time



1. Welcome (4min)
2. Questions to the user (3min)
3. Give context (3min)
4. User tries feature (30min)
5. Summary and questions (5min)



this book explains the process very well

dig - draw - prototype - test

Make the users tell **out loud
what they're thinking.**

**Focus on what they don't
understand.**

dig - draw - prototype - test

Outsource the test
if you can't spend a day organizing it

USERBRAIN &  UsabilityHub

are two cool solutions

But don't skip it!

dig - draw - prototype - test



Make a list of the **top 5** problems users encountered.

dig - draw - prototype - test



**Go back to the draw
step to fix them.**

a message that explains X

a message that explains Y

an arrow to show the user

**Try not to add things
but remove what makes your
feature complex instead.**

another option

a message that explains Z

**Try not to add things
but remove what makes your
feature complex instead.**

Repeat loops

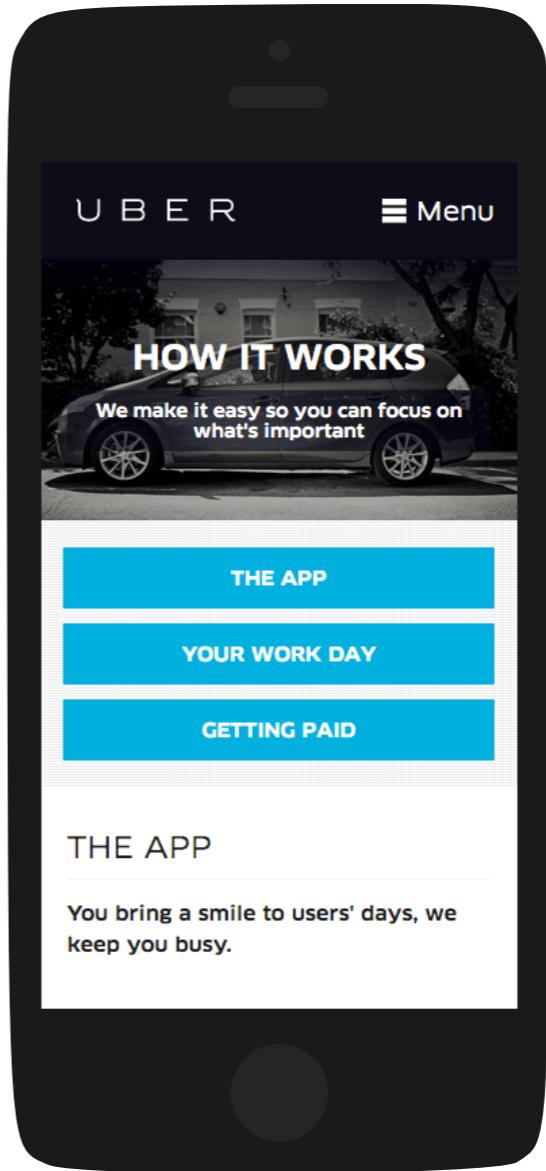
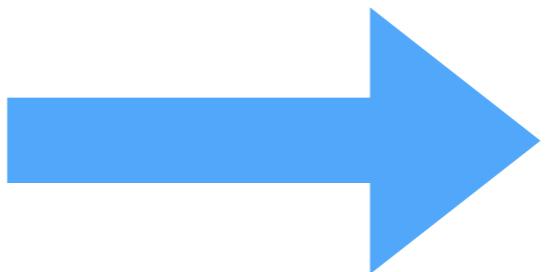
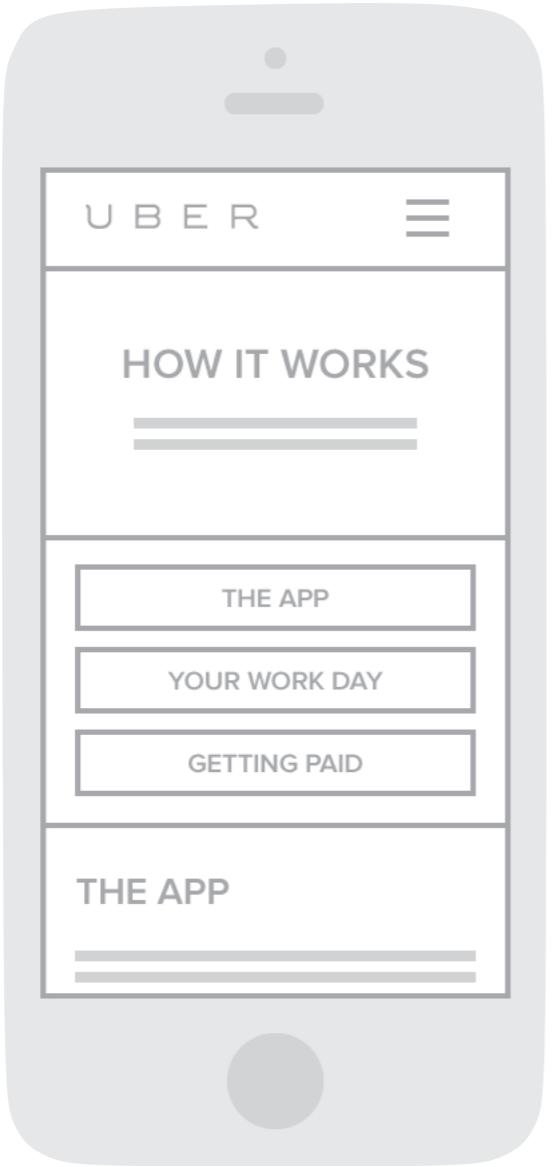
until your feature is easy to
use and people see value in it



UI and code

That's where you need
a good **UI** designer
and some talented
developers





Make the design **pretty** and code a
clean version of the feature!



Roll-out and Analyse

Roll-out your feature gradually

first to 1% of users

then to 5%

then to 20%

then to 50%

finally to 100%

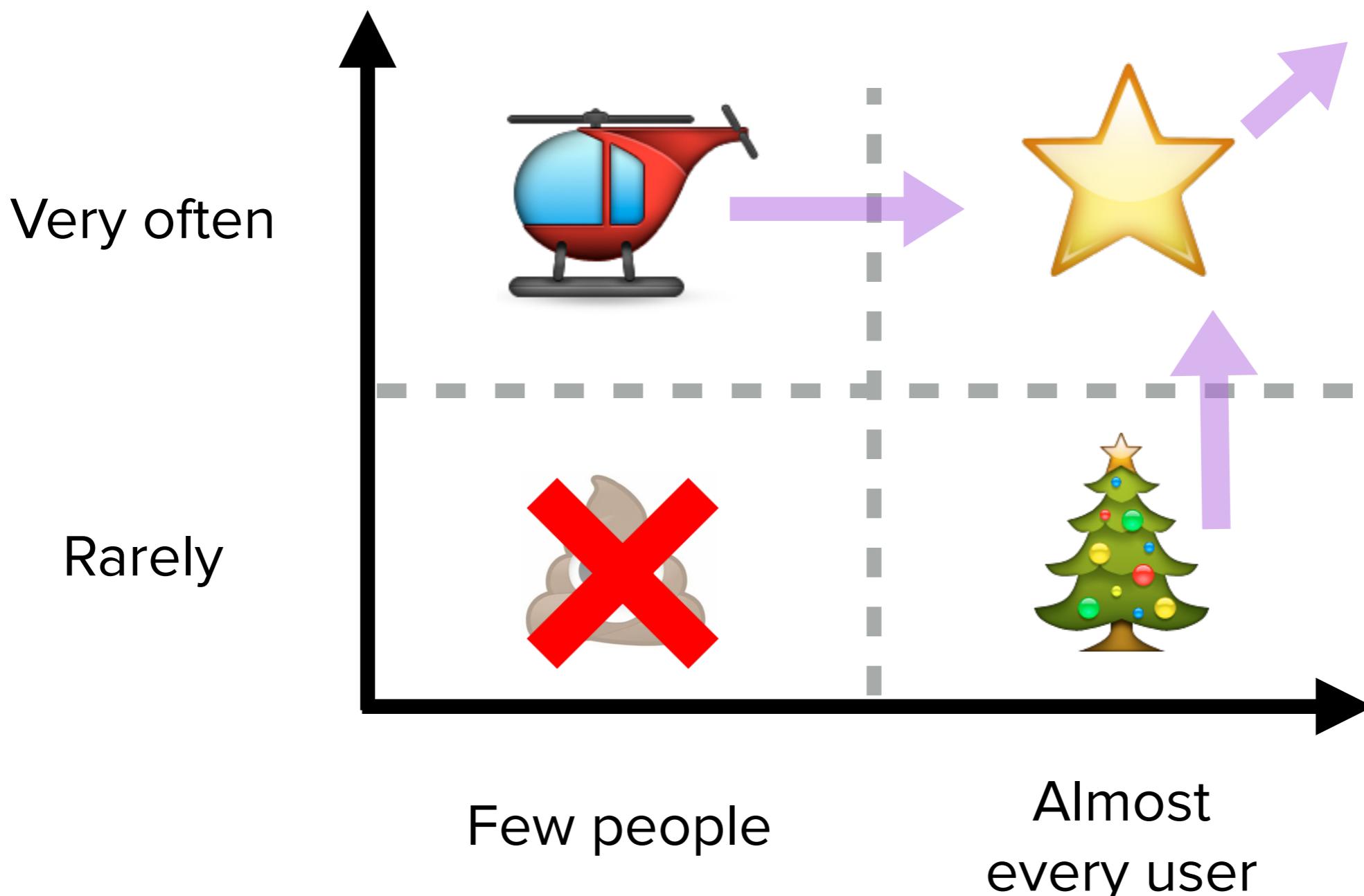


Tell your developers to visit featureflags.io to see how to do it

A scenic view of Milford Sound, New Zealand, featuring towering green mountains and a deep blue fjord.

**Facebook tests most new features in New Zealand
(2M users) before rolling out globally.**

Features' usage should go like this



**Roll out to
more users**

every week, if performance is good

Check your NPS and Retention

Start Day	Users	DAY 0	DAY 1	DAY 2	DAY 3	DAY4
Jul 22	12.233	100%				
Jul 21	13.552	100%	57.6%			
Jul 22	9.341	100%	60.4%	40.9%		
Jul 19	6.419	100%	43.0%	42.8%	31.1%	
Jul 18	7.435	100%	28.0%	28.6%	29.9%	22.4%
Jul 17	12.467	100%	32.4%	19.2%	21.3%	23.3%
Jul 16	11.535	100%	57.0%	23.6%	14.7%	18.0%

It's not a success if the overall performance of the product is worse !



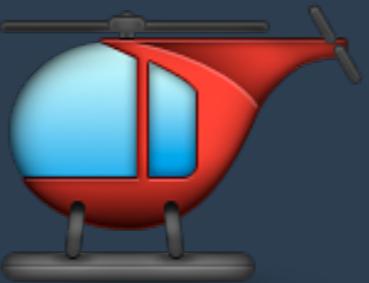
Congratulations!

Your product made a big step towards greatness

Do it again for another feature



or



or



Remember the process

1. Understand your users
2. Select a feature
3. Design loops!
4. UI and code
5. Roll-out and analyse

These books and blogs are great:

Intercom on
Product Management



 Intercom

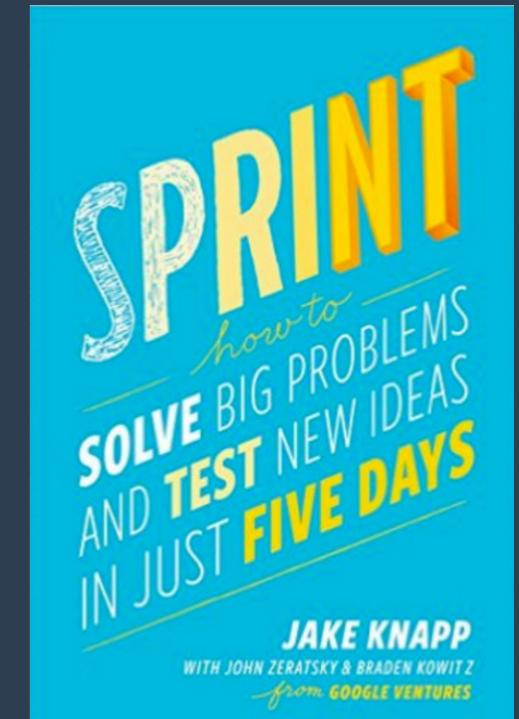


The Hanno Playbook

This is the way we sail our ship.

YC
Startup Class

CSS Wizardy



I used them to write these slides :)

THANK YOU

wanna talk about your product ?



Maxime Braud

@mxbraud

mx.braud@gmail.com

