



Building Data Driven Products... At Scale

Nikhil Dwarakanath





In the last 18 months...

**5 Years to 1
Billion Rides**



**9 Months
thereafter to
our next Billion**

**Predominantly
Transport**



**Transport
Food
Deliver
Pay
Et Al**

**68 MM
Downloads**



**125 MM
Downloads**



In the next 30 mins

Set context of the problem

Share how we solve it

Showcase an example

What usually happens with Early Stage Products



- Product Managers & Engineers think up **cool** stuff
- A development cycle is then **quickly** created around these features
- Built features are then taken to **production**

...The Fallout



- Less important product features get picked up
No objective prioritization, besides the PM's clout
- Features are lauded as successes immediately following the launch
Makers usually Checkers
- [Some] of these same features may then be quickly switched off / rolled back / bandaged
No tangible impact near term | Success not defined well upfront

This is all the more difficult for us...



- 8 Countries | 235+ Cities
- 2 Billion + Rides
- 125MM + Downloads



- The Yin & Yang of the network
- Core Human Need
- Same, yet different
- 20 TB of Data processed everyday

Therefore, imperative to view the Product with a Data lens



- Comprehend **user interaction & product performance** data
- Develop **Data driven Hypotheses**....
And therefore Feature Development
- **Prototype** at Massive Scale

This needs Product, Engineering & Data teams to work across the Dev cycle



How do we *typically* do this
^

1. Instrument...like crazy

Front End | Back End | System

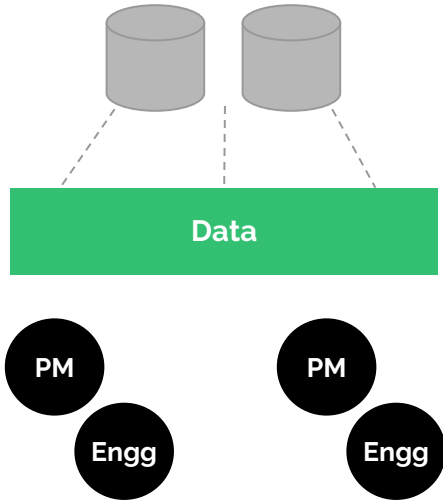


Track!

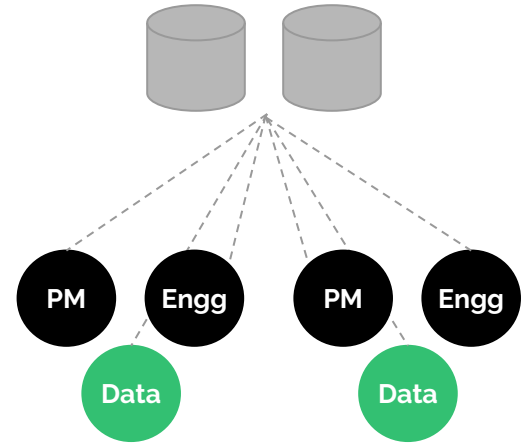
Track!

Track

2. Integrate & democratize scalable systems to mine data



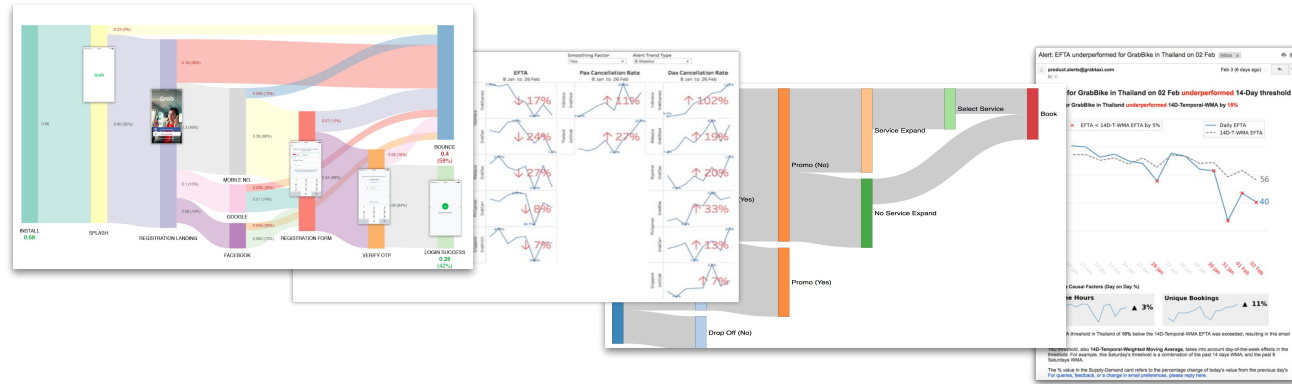
Usually



Today

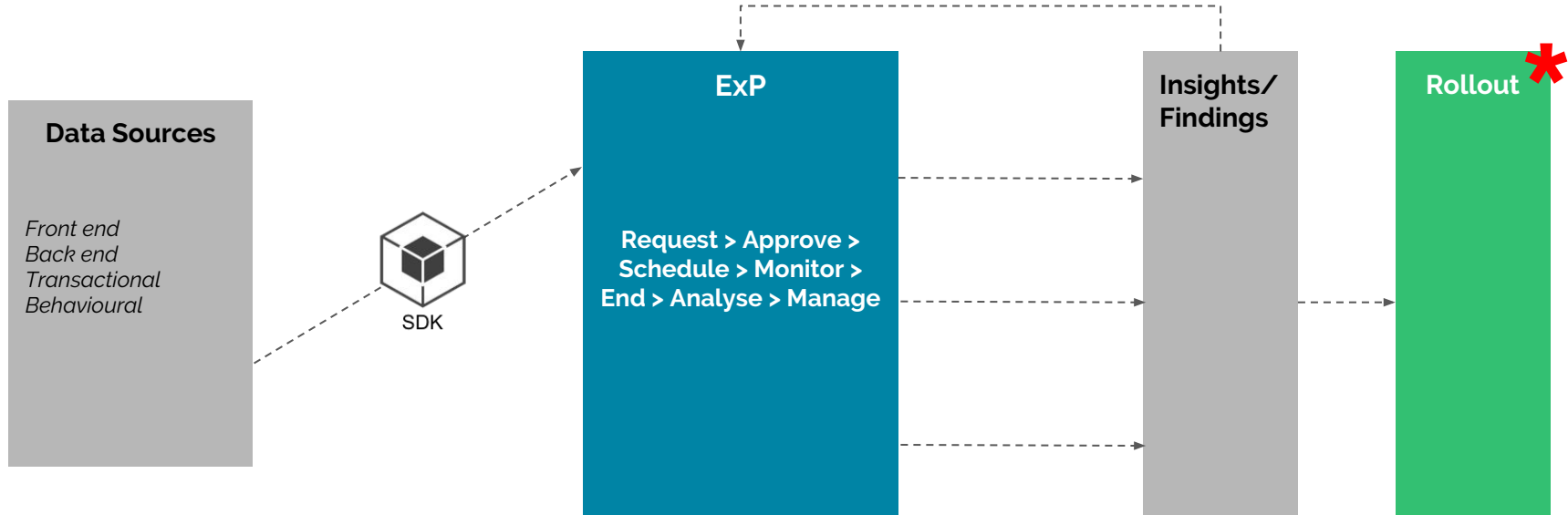
3. Generate (a large # of) data driven hypotheses

- A.** Define Target Metrics & Set Ground Rules
- B.** Build Heat maps, Information systems, etc., → Identify anomalies, causality, opportunity areas



- ### C. Build hypotheses for plausible features

4. [A | B] Test!



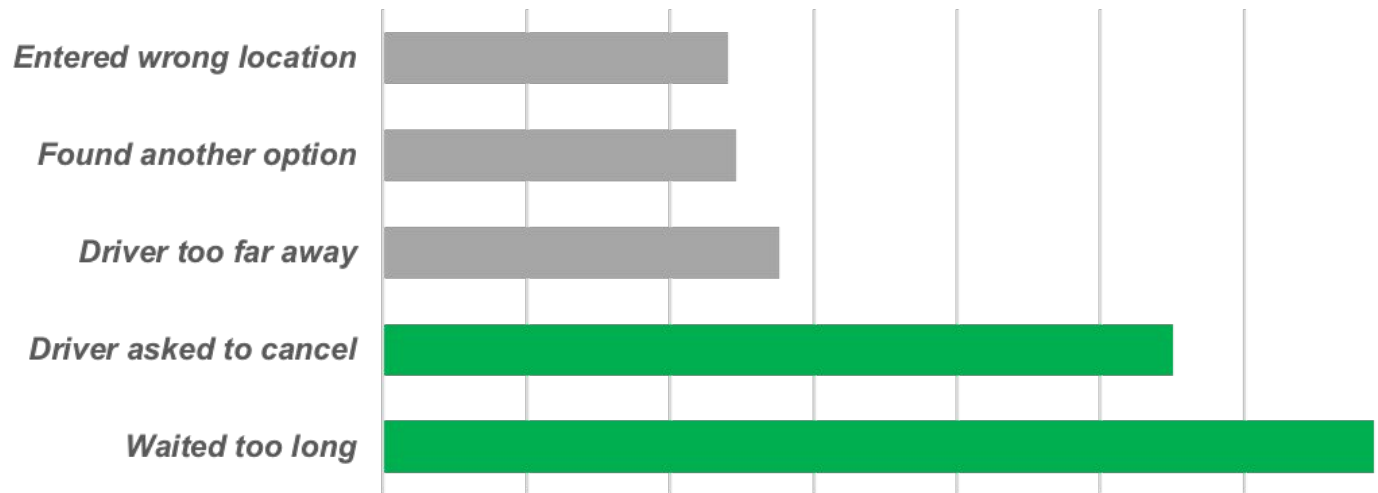
- Democratized testing across product Families
- Standardized & partially automated measurement
- Continuous effort to make it easy to iterate & rollout



What does this ^{usually} translate into, for our users

Circa 2017

High #cancels.... after 2 mins. of a vehicle getting allocated

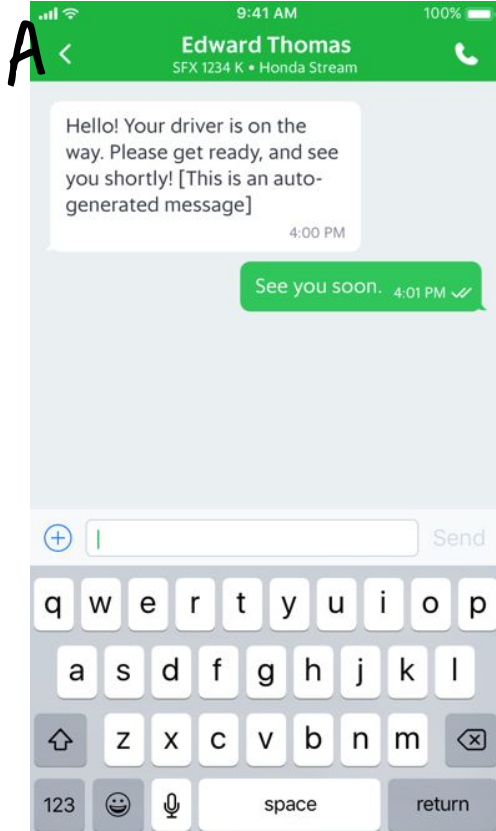


A majority of these
were attributed to
the driver &
passenger not
finding each other



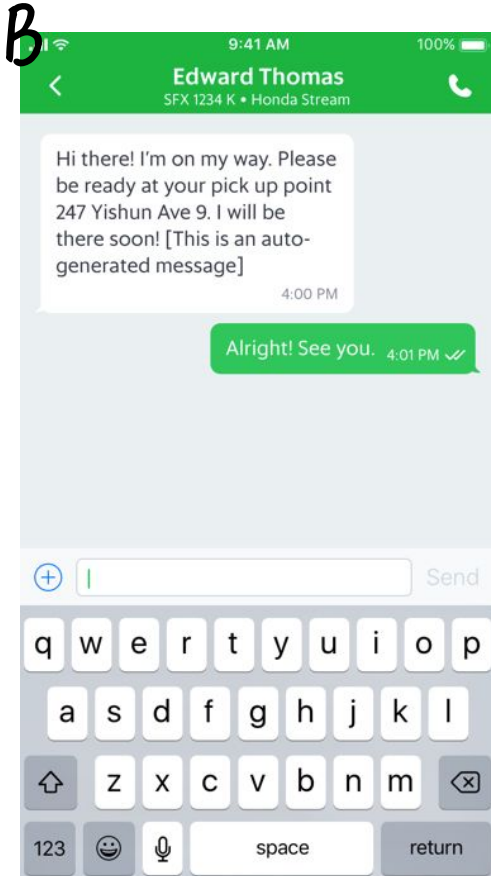
So, Allocation has happened and
Intent has been expressed... *by both parties*

We tried Native chat



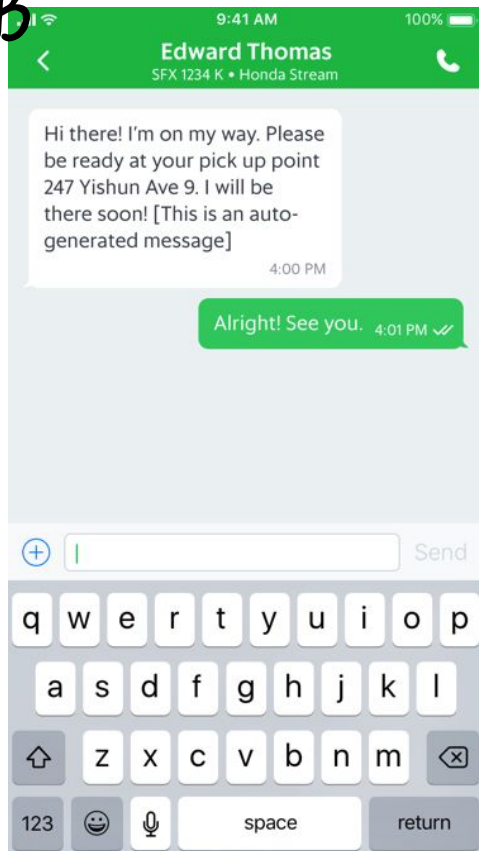
- A|B tested an **informal line of communication**
- Seldom a **real** conversation
- **But guess what!!**

On my way...



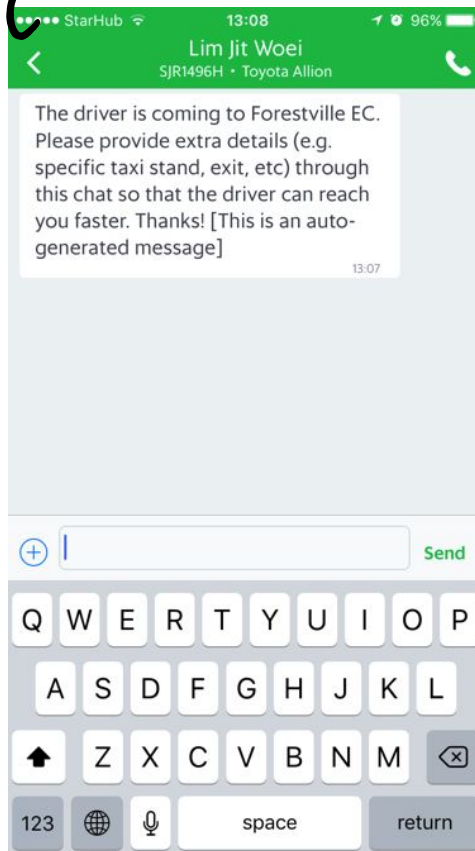
On my way...

B

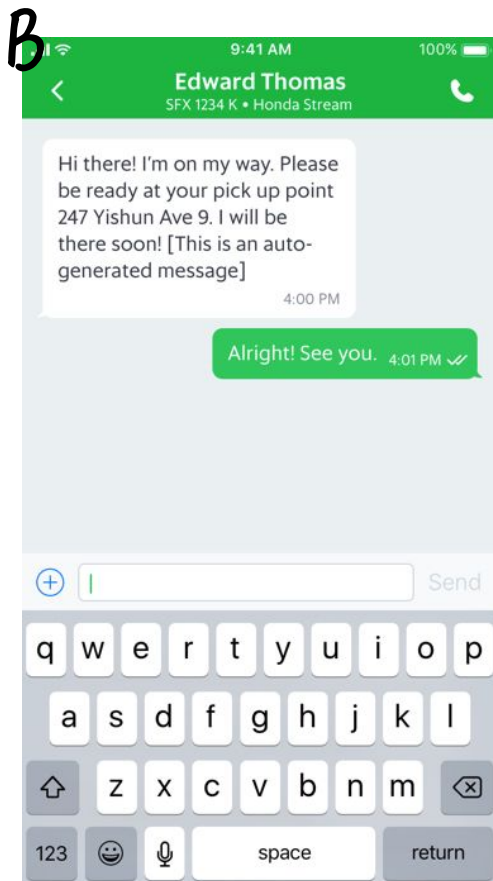


Where are you...

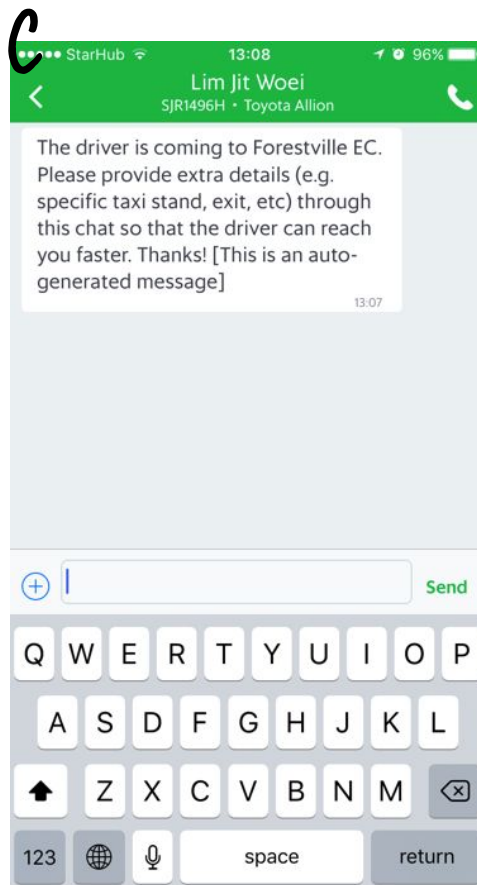
C



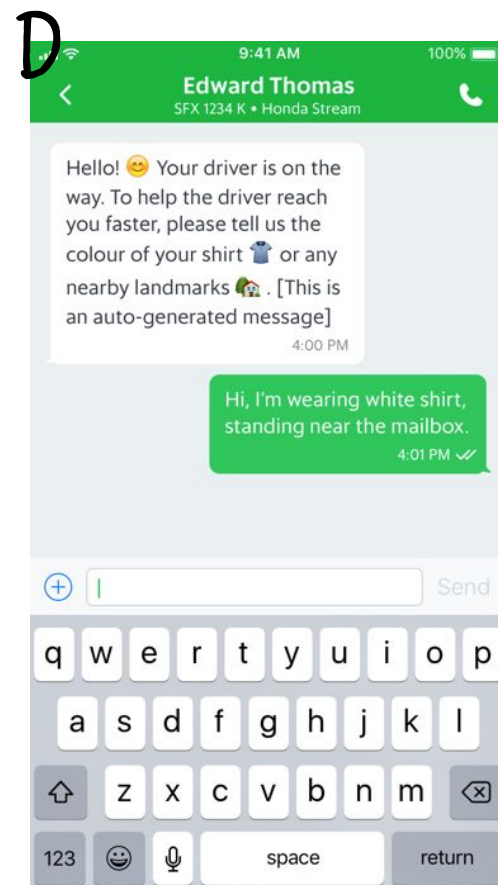
On my way...



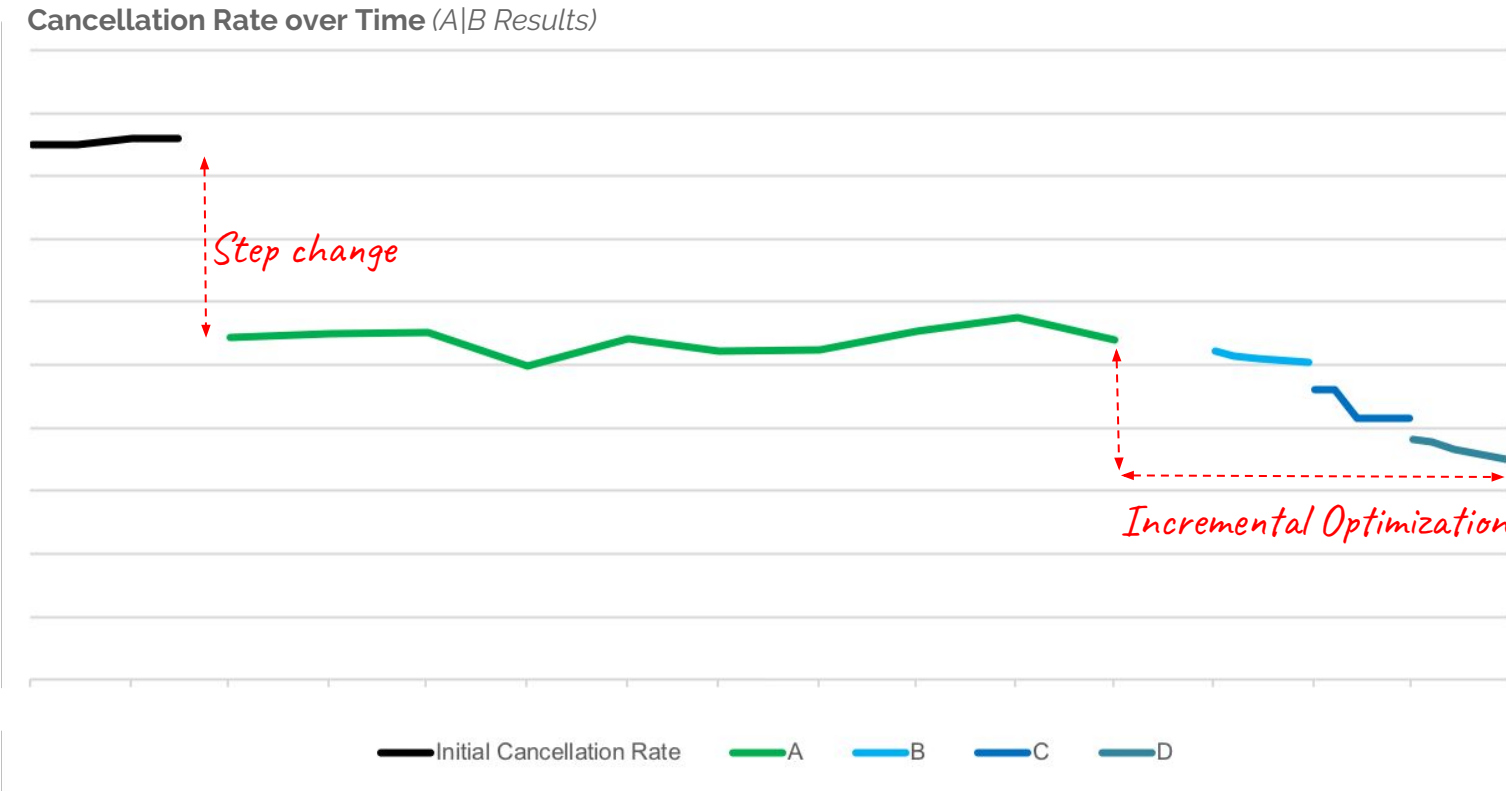
Where are you...



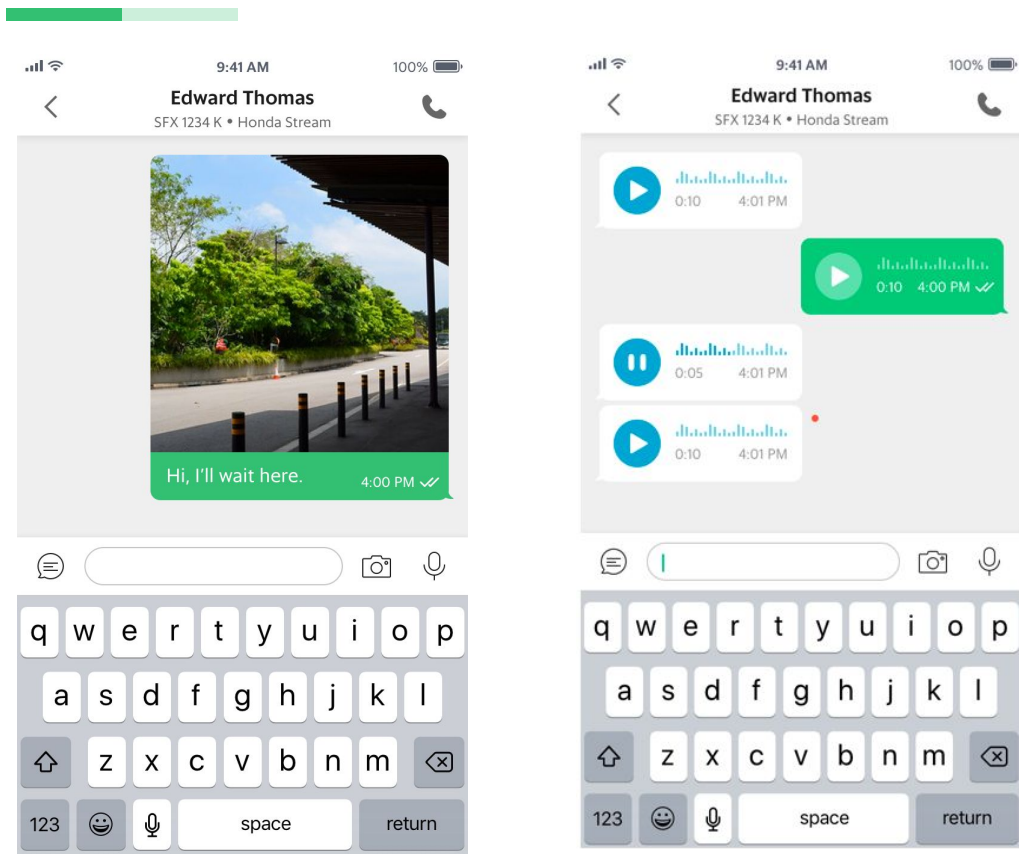
What are you wearing...



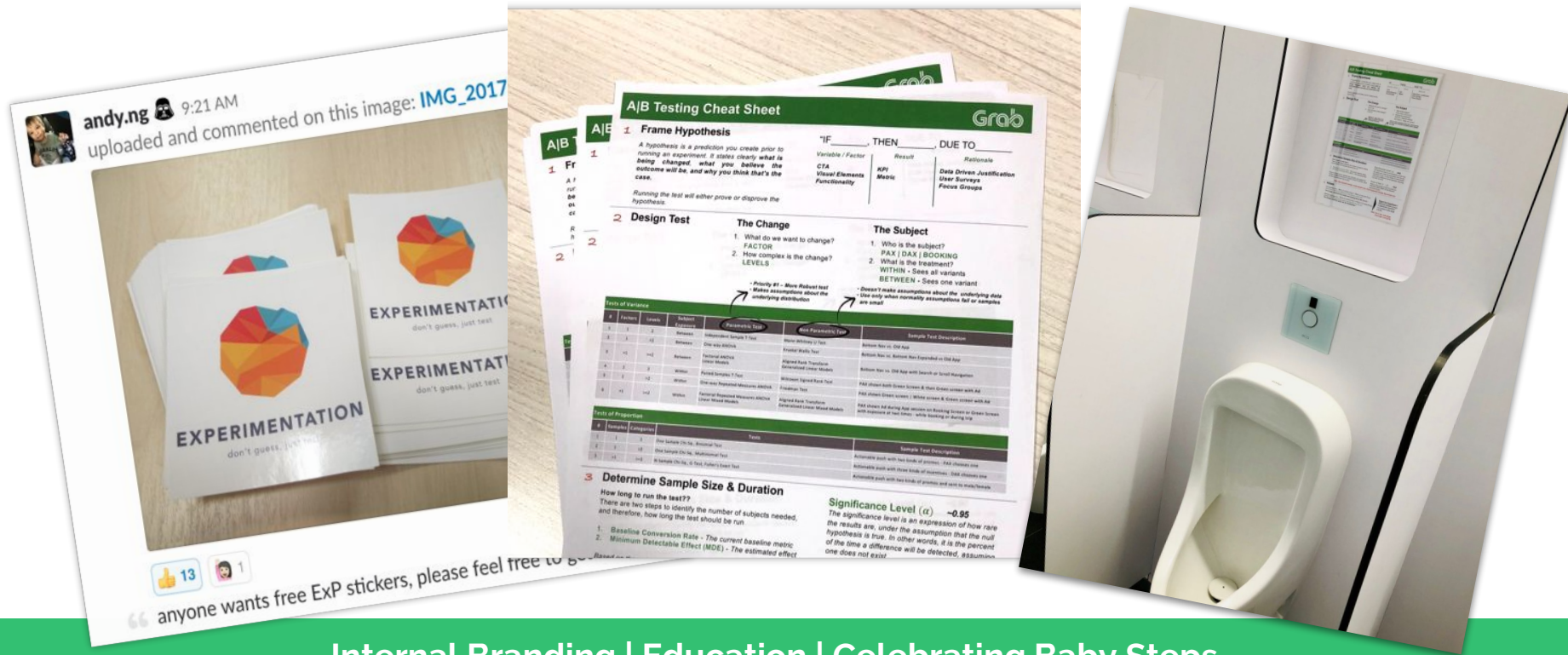
The Outcome



Today,....



Building this ethos has meant...



Internal Branding | Education | Celebrating Baby Steps...



Is there a downside?



Absolutely!

Consider what is the cost of the average test...



Engineering:

SDK Integration

PM / Analytics:

Test Setup

Design:

Variant Creation

QA:

Performance Testing

Analytics:

Monitoring, Measurement & Communication

Assuming a test runs for ~2 weeks....

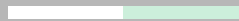
++ Delay to Market ~2 weeks or more

When not to Test



- Avoid simple CTA type tests
- When we didn't have something and now do
- When the test is a proxy for something else

That said, test everything ~~else~~ possible



Thank You.

And We Are Hiring!!