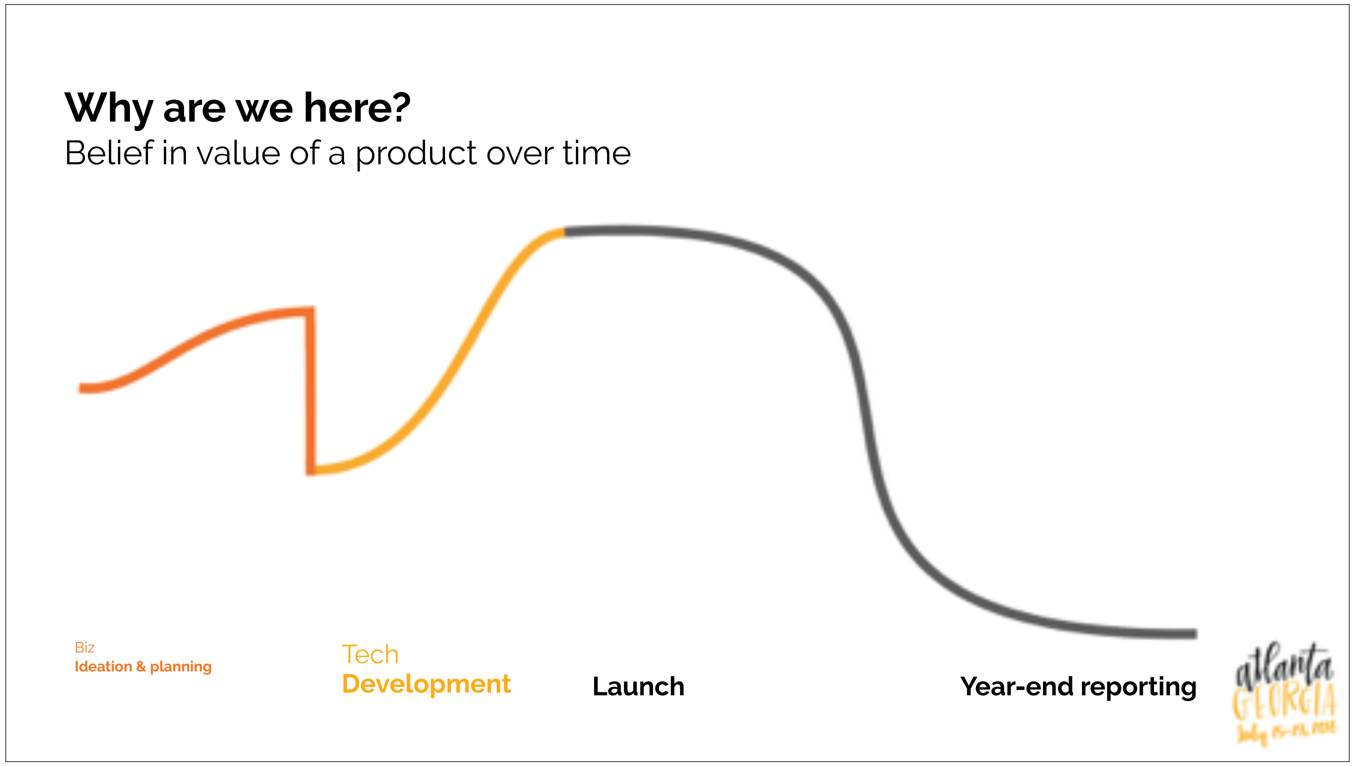
Fortune teller to scientist

a lean approach to predicting successful products

Kylie Castellaw, Hugo Corbucci, Ashley Pandya
ThoughtWorks



Why are we here?

Belief in value of a product over time

Well let's look at how a 'typical' tech project goes down…

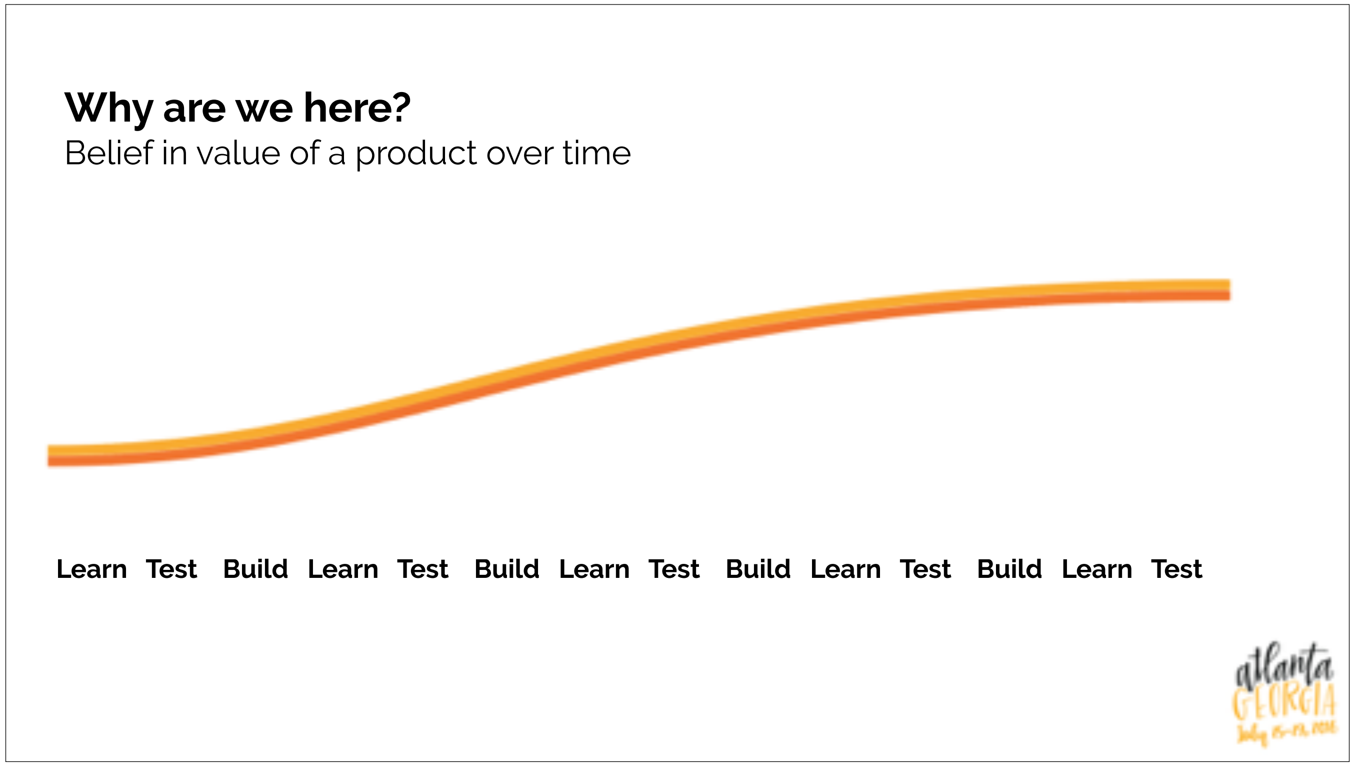
A business team comes up with an idea and plans a project. They convince themselves it's a great idea and it will be a slam dunk -- if only the technical team can implement it well.

Then the project is handed off to a technical team to implement. The technical team has little understanding of why this project is happening, and doubt the whole thing. But as they start building it, they become proud of the work they've done and start to believe it will be a success in the marketplace as well.

Then after some time -- up to multiple years of development later -- the completely developed solution is released into the wild. Now's when the money will start rolling in!

Little measurement is done until perhaps at the end of a quarter or even the year. At that point the numbers aren't hitting the forecast and it seems like the whole thing might have been a bad idea on the business's part -- or that's what the tech team thinks. The business team thinks it must not have been implemented well; there are some bugs and not all the features they asked for made the cut to be developed.

What happened?

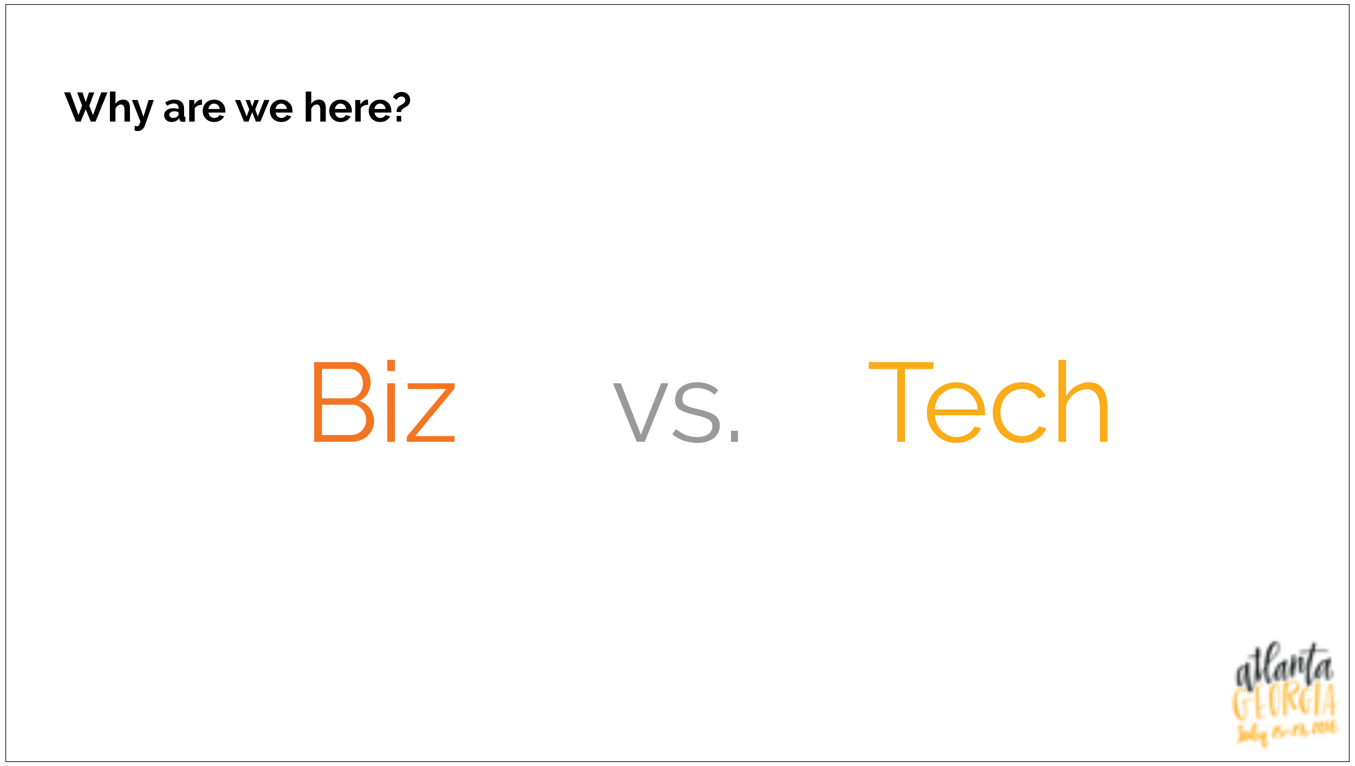


Why are we here?

Belief in value of a product over time

Learn Test Build Learn Test Build Learn Test Build Learn Test Build Learn Test

We want a slow and steady evolution where we always feel confident about the results we're going to get



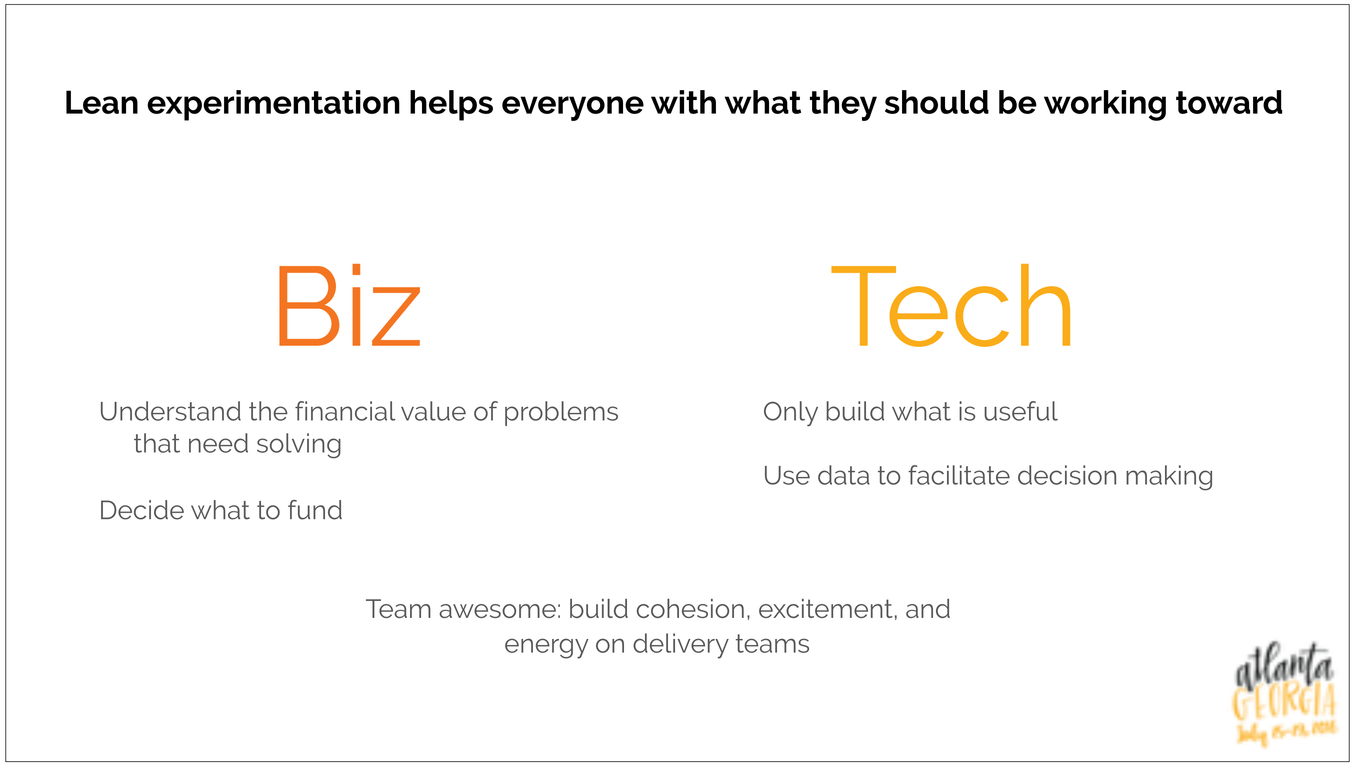
Why are we here?

Biz vs. Tech

Additionally, -- and surprisingly relatedly -- it is a fact universally acknowledged that business and technology teams are rarely on the same page, and lean experimentation can help with that too.

Ever gave your blood, sweat, and tears to a project only to have it slowly die or be cancelled after launch because it wasn't getting the anticipated traction in the marketplace?

Ever had or witnessed a clash on a project between technology and business groups? Why is that so common?



Lean experimentation helps everyone with what they should be working toward

Biz

Understand the financial value of problems that need solving

Decide what to fund

Tech

Only build what is useful

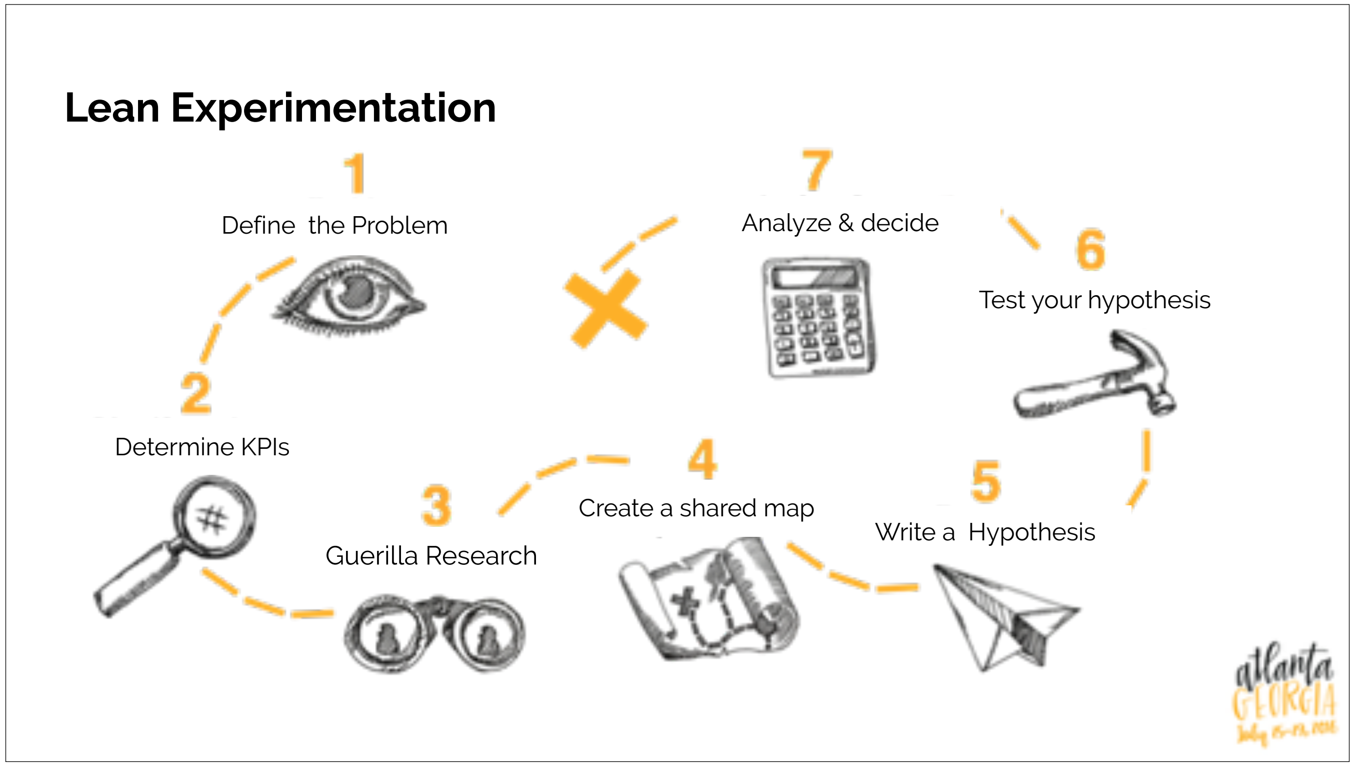
Use data to facilitate decision making

Team awesome: build cohesion, excitement, and energy on delivery teams

Well biz and technology inherently care about different things. After applying lean experimentation very successfully in an interdisciplinary environment, we figured out why it helps bridge this business-technology divide: it helps both groups do the things that they care deeply about -- and have a shared language for it. All of a sudden, they have a shared vision of the product and an evolving understanding of how much value to expect. They pivot together when necessary, and launch incrementally with much less risk and friction.

In constantly validating assumptions, re-evaluating the path forward, and testing with users along the way, the final product is less likely to flop in the market - and less likely to take years to launch, avoiding the pitfalls of launching a product that may be completely irrelevant to the new market conditions.

It also bears mentioning that though we aren't focusing on design teams for the purposes of this talk, some of designers' traits are similar to those of business people, and some are similar to technology teams. Therefore lean experimentation is a natural fit for designers, and often they can be great allies in implementing and leading a lean experimentation effort.



Lean Experimentation

1 Define the Problem

2 Determine KPIs

3 Guerilla Research

4 Create a shared map

5 Write a Hypothesis

6 Test your hypothesis

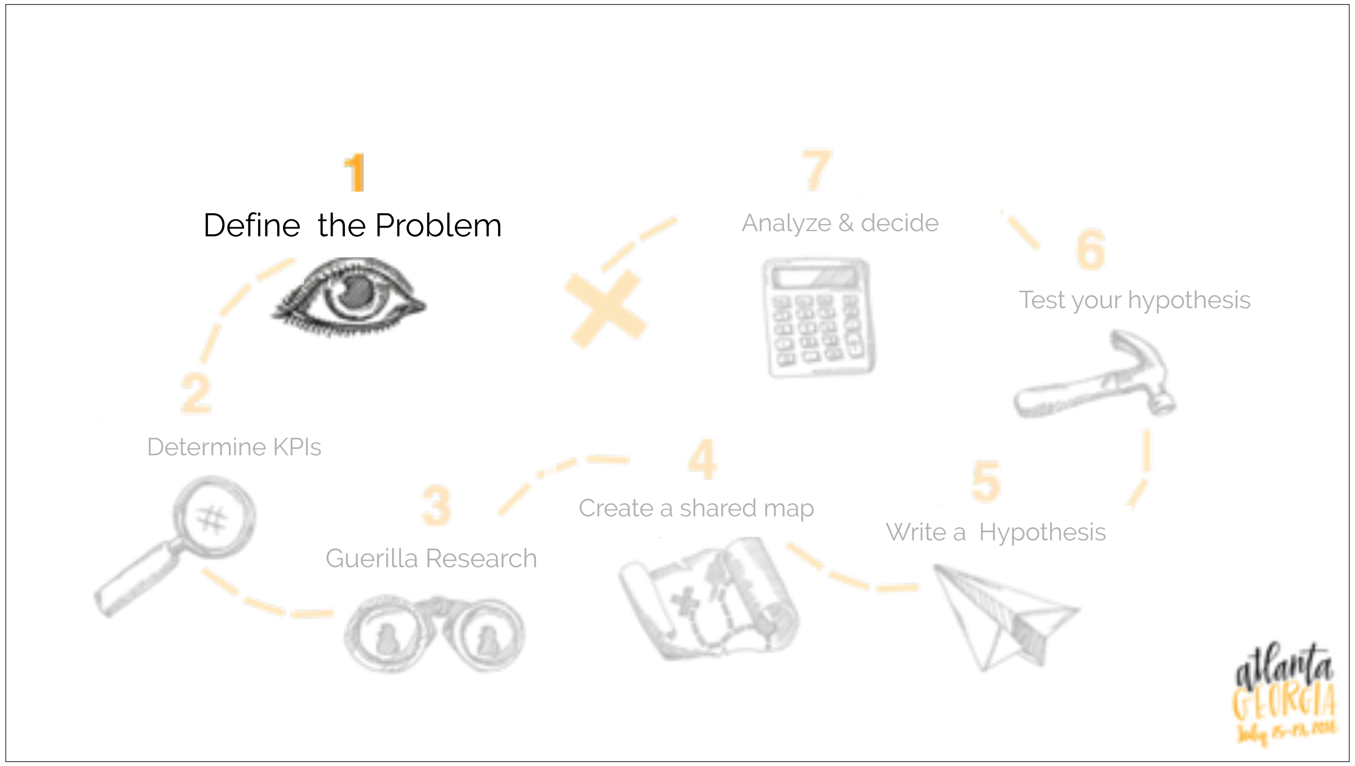
7 Analyze & decide

Today we’re going to talk about a lean approach to product experimentation. The goal is to familiarize you with the process and try it out yourself so that by the end you’ll be ready to put lean experimentation to use on your next project!

The process is an iterative and ongoing one, but can be broken down into 7 steps, and we’ll go through it all:

1) We’ll start out by Defining the Problem…
2) Then we determine the best KPIs/Metrics to measure success…
3) Next is Learning through Guerilla research…
4) Followed by Creating a map with focus areas…
5) Then we’ll cover how to write a testable Hypothesis…
6) … and test it
7) … and understand the impact of the work you’re doing.

The process is an iterative, loopy one, but by understanding these 7 basic steps you’ll have the building blocks to need to put Lean Experimentation into practice yourself.



1 Define the Problem

2 Determine KPIs

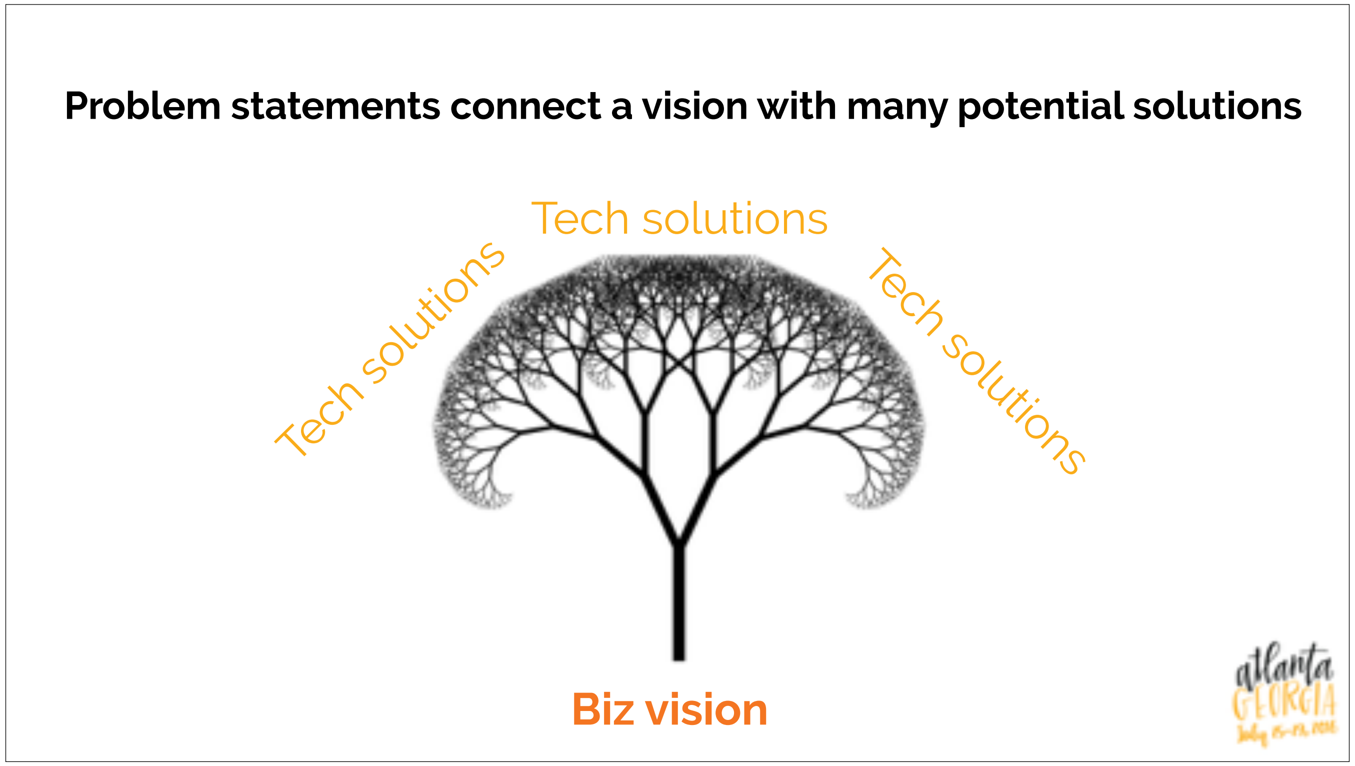
3 Guerilla Research

4 Create a shared map

5 Write a Hypothesis

6 Test your hypothesis

7 Analyze & decide



Problem statements connect a vision with many potential solutions

Why are we doing this project? What is the future we envision? What is the problem we are trying to solve?

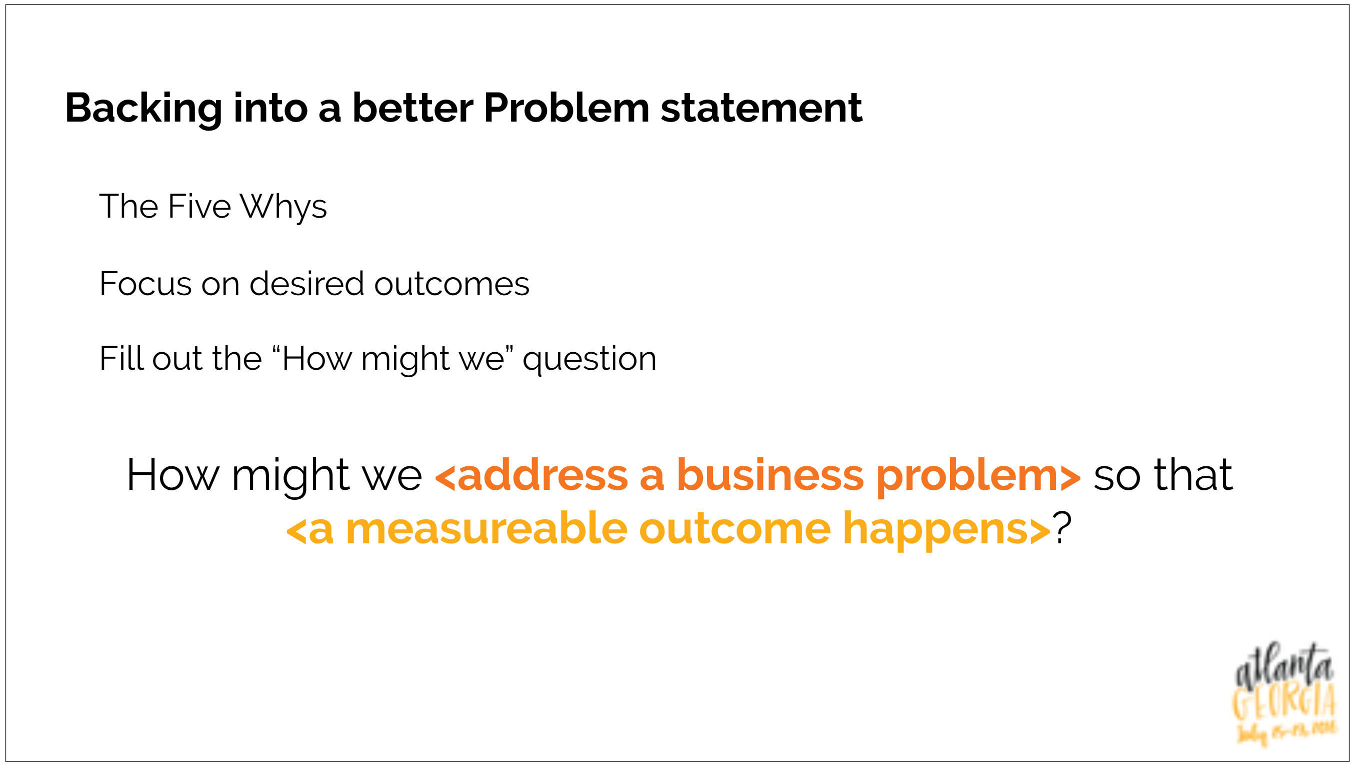
Sometimes you are given a project directive, goal, or ‘problem statement’ that is a solution, not a problem. But for endeavors that are going to follow the lean experimentation method -- or any project where you want buy in/collaboration from contributors -- you’ll need a better one.

A good problem statement is GENERATIVE instead of PRESCRIPTIVE: it allows the team to think creatively about solutions instead of predetermining the solution.

Backing into a better problem statement

“We need an app for waiters to take orders at a restaurant.”

Here’s an example of a PRESCRIPTIVE problem statement: ”We need an app for waiters to take orders at a restaurant.”.
Why is this bad? (Ask audience…) It’s because this specifically tells the team what they need to build, instead of allowing them to be creative in finding a solution to the problem.



Backing into a better Problem statement

The Five Whys

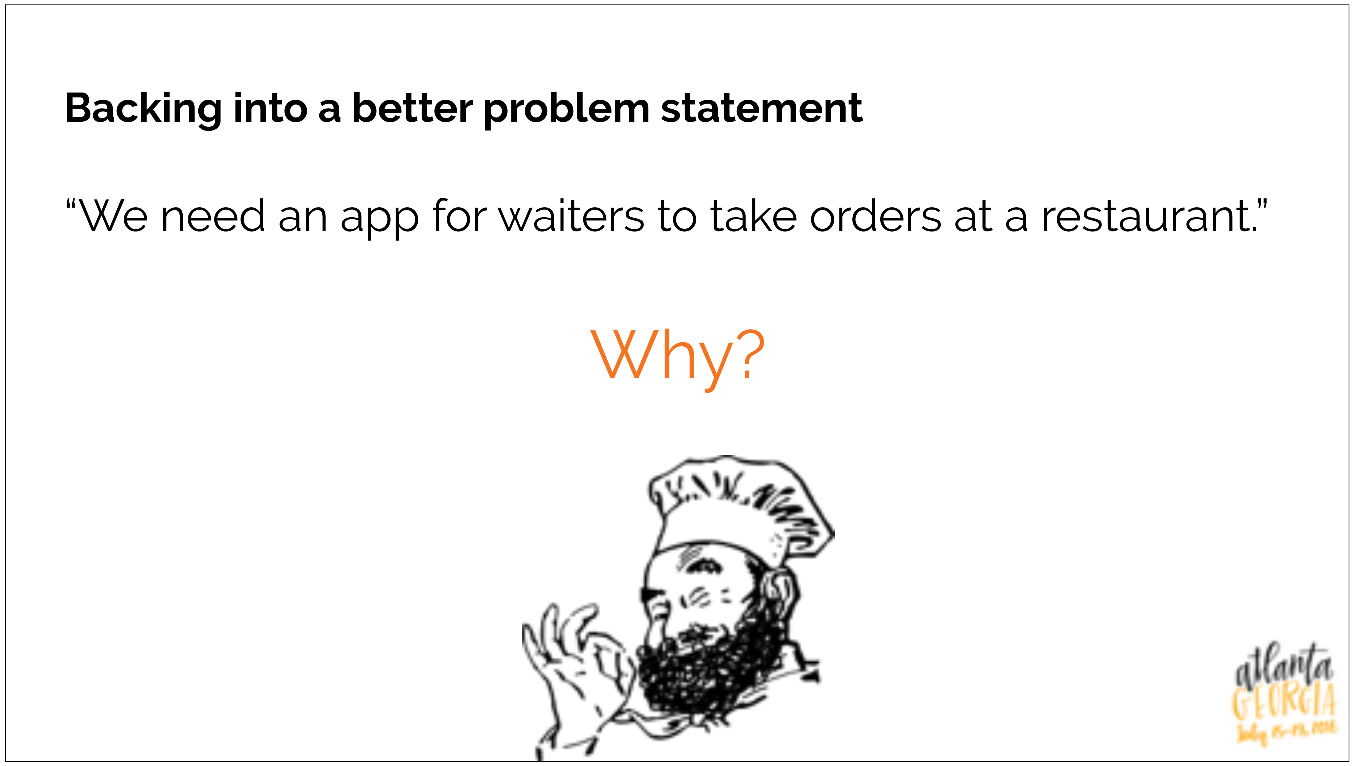
Focus on desired outcomes

Fill out the “How might we” question

How might we **<address a business problem>** so that **<a measureable outcome happens>**?

A few techniques exist to help finding a better problem statement.

The 5 whys is one of the most famous ones.
We like to try to fill this How might we template.

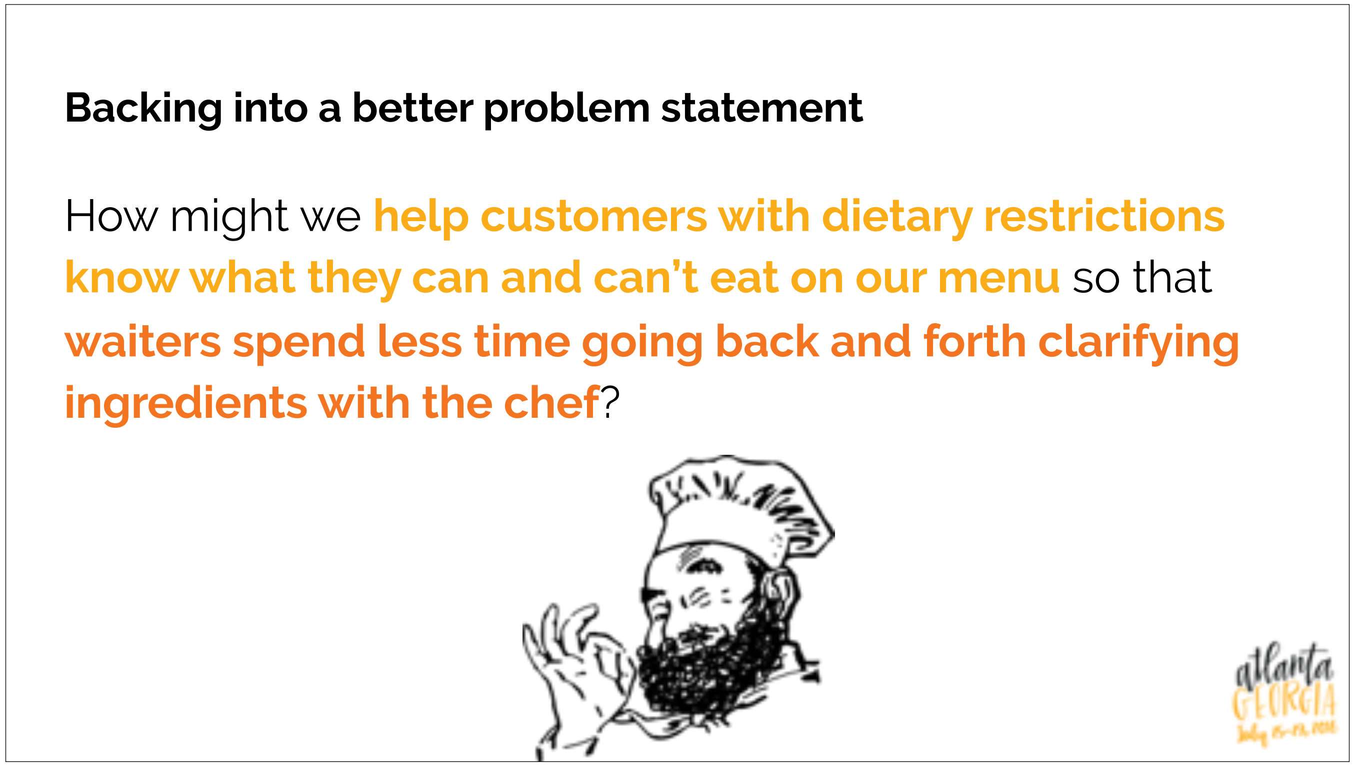


Backing into a better problem statement

“We need an app for waiters to take orders at a restaurant.”

Why?

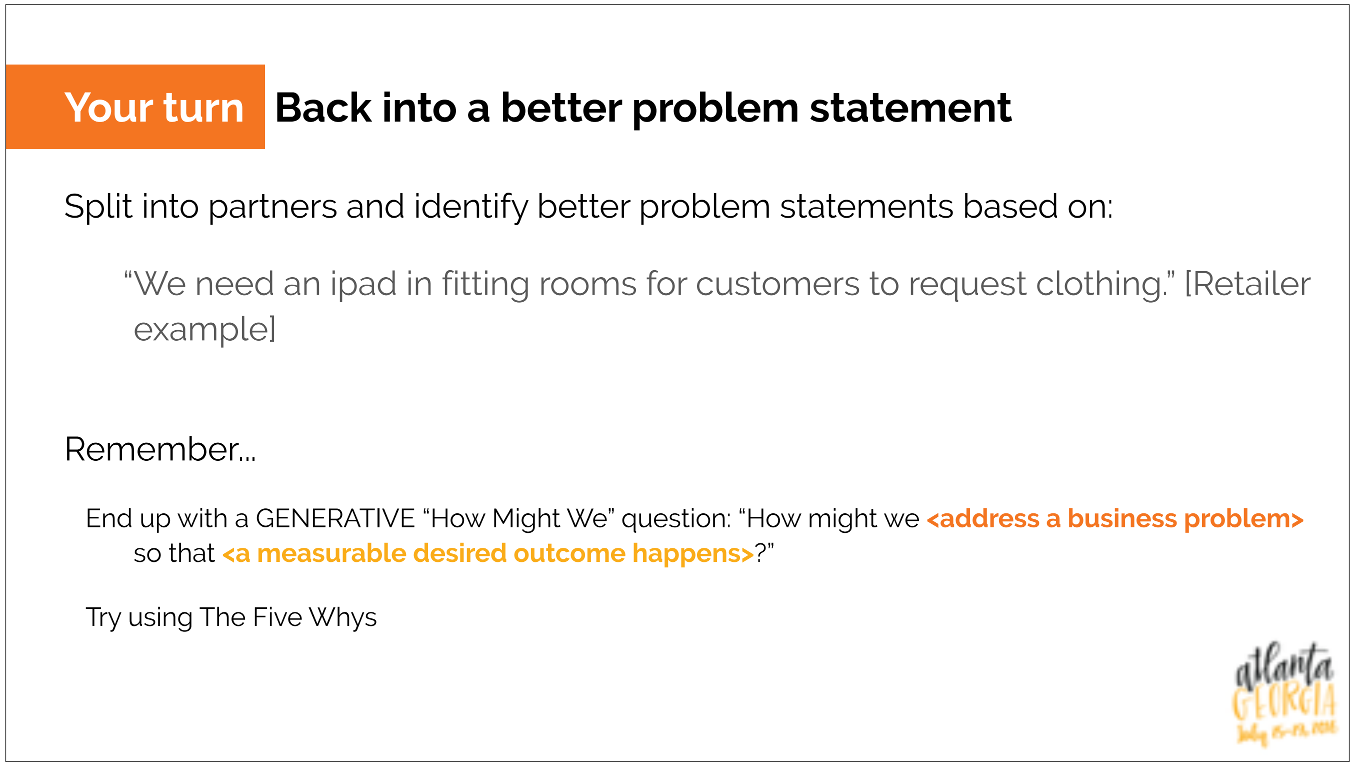
Example: WHY?
Because waiters spend a lot of time going back and forth between the kitchen and the customers asking menu questions. WHY?
Because customers are asking a lot of questions about ingredients. WHY?
Because customers have increasing dietary restrictions and sensitivities/allergies. WHY?
Because the menu changes very frequently and waiters can’t remember ingredients of new recipes.



Backing into a better problem statement

How might we **help customers with dietary restrictions know what they can and can't eat on our menu** so that **waiters spend less time going back and forth clarifying ingredients with the chef**?

Now we know what is the change we want to see and the impact we expect from it.



Your turn Back into a better problem statement

Split into partners and identify better problem statements based on:

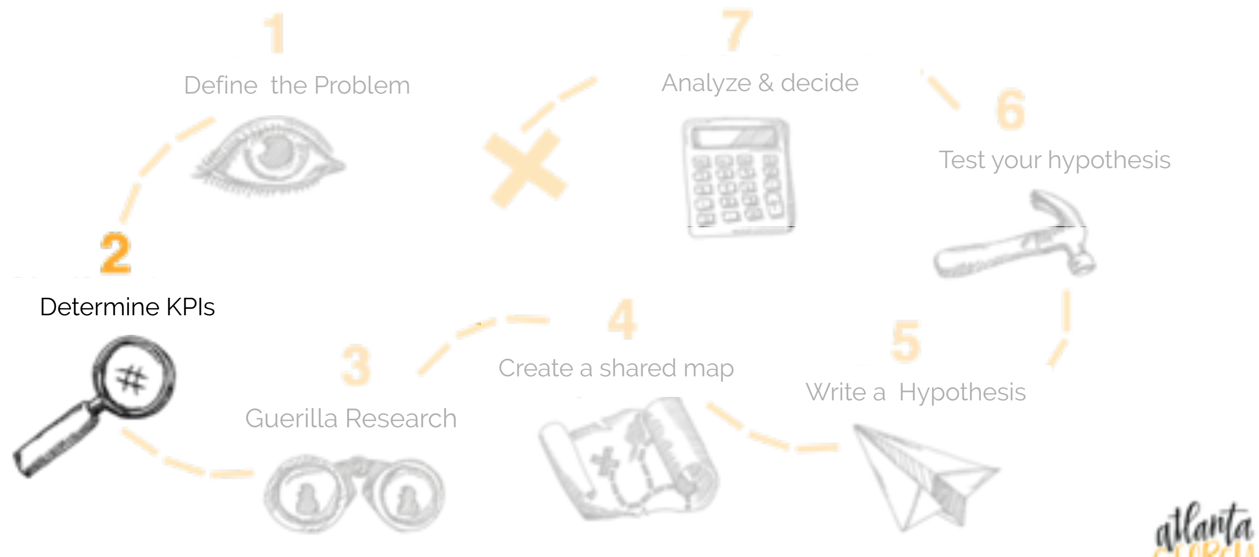
> “We need an ipad in fitting rooms for customers to request clothing.” [Retailer example]

Remember…

End up with a GENERATIVE “How Might We” question: “How might we **<address a business problem>** so that **<a measurable desired outcome happens>**?”

Try using The Five Whys

One partner takes the “bad” problem statement and pretends to be a business stakeholder for that company. The other partner then uses the 5 Whys to find the real problem. Then write a better problem statement together. You’ll have just 5 minutes total.



1
Define the Problem

2
Determine KPIs

3
Guerilla Research

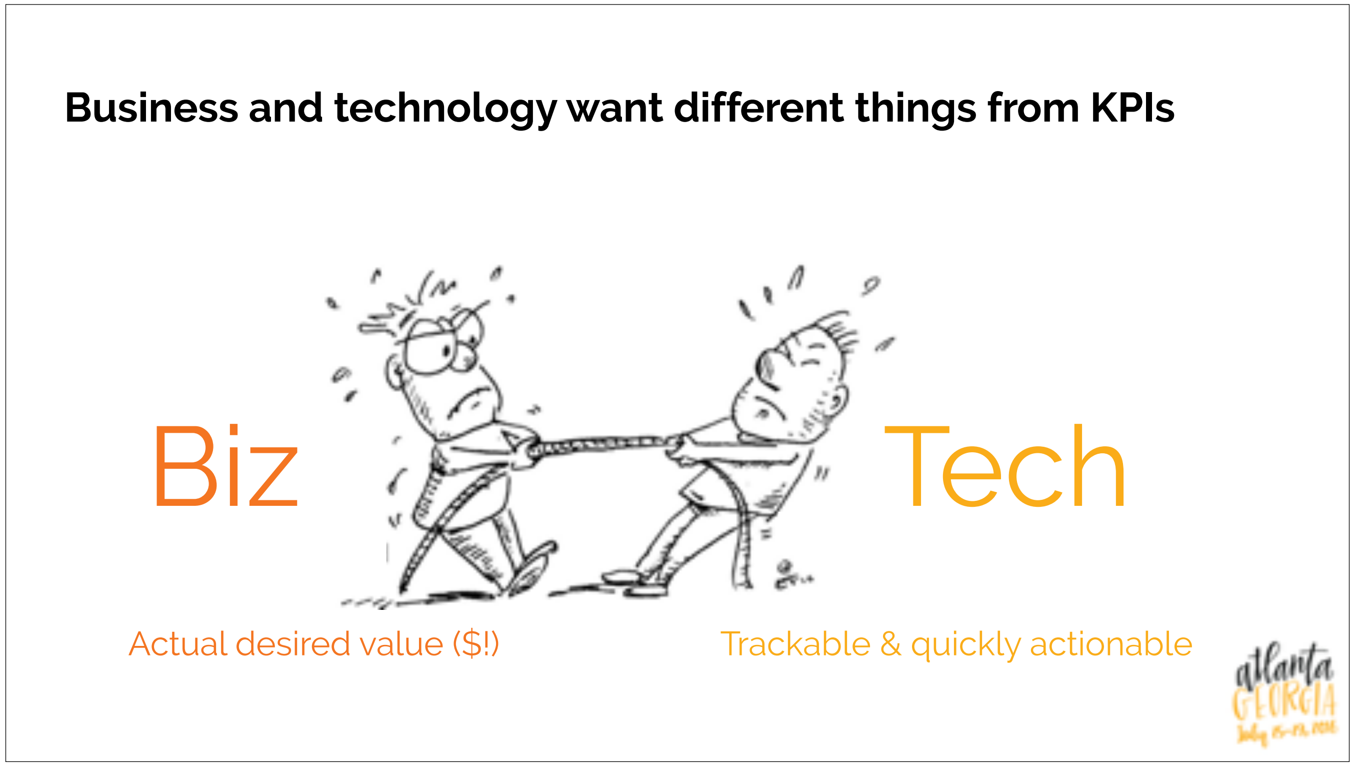
4
Create a shared map

5
Write a Hypothesis

6
Test your hypothesis

7
Analyze & decide

atlanta
GEORGIA
July 15-17, 2016



Business and technology want different things from KPIs

Biz

Actual desired value ($!)

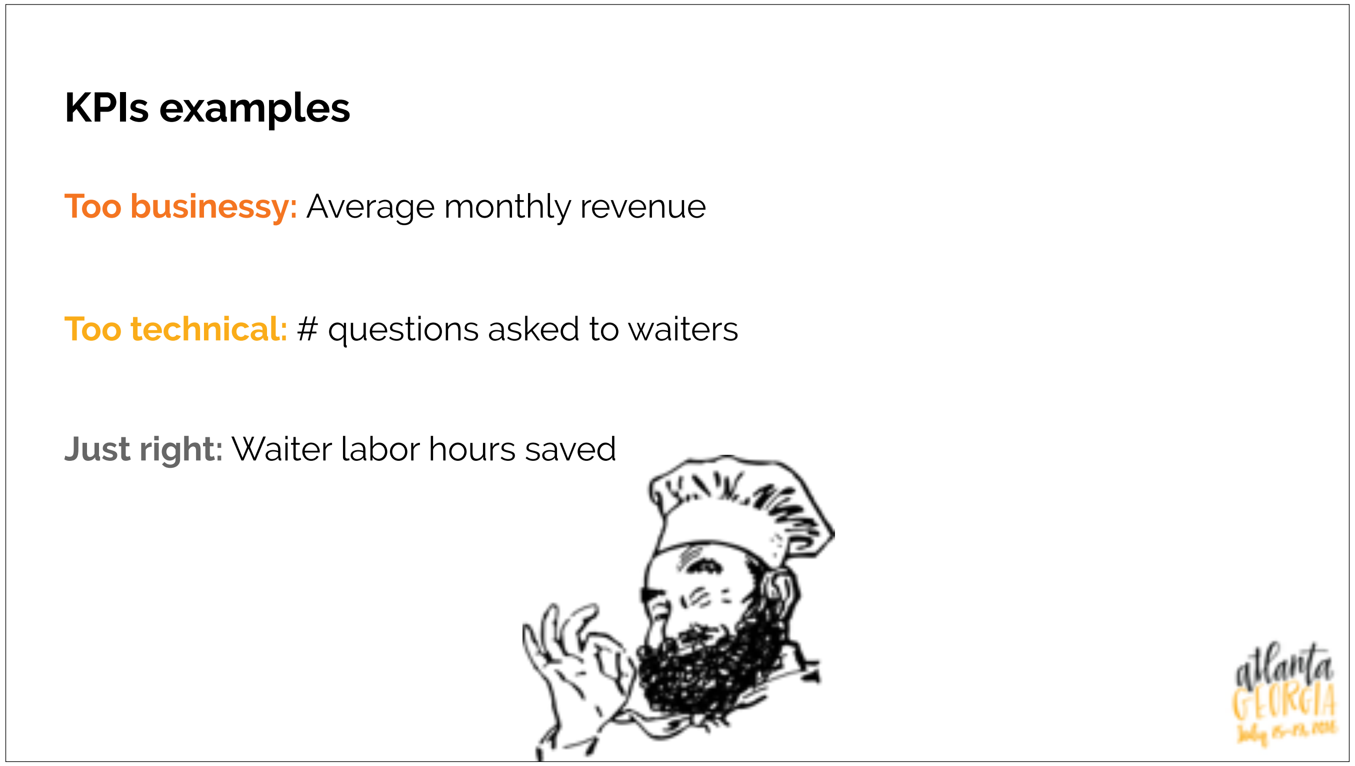
Tech

Trackable & quickly actionable

How does your idea drive business or customer value? All of your experiments should aim to affect this one variable, or the “one metric that matters.” Everyone can probably agree that being able to measure value delivered from a project is a good thing. The tricky thing is agreeing on a KPI/Metric that everyone can get behind.

If you’ve ever been in a room with business and technology people trying to agree on what’s important to measure, the business folks unsurprisingly want to measure things like revenue; they care about metrics that are directly related to the bottom line. Technology people, on the other hand, want to see metrics that they can track with as little effort as possible and are quickly actionable. They are, after all, people that thrive on solving problems.

This is a tug of war, but the important thing to remember is that no one wins if the rope is pulled too far to either side.



KPIs examples

Too businessy: Average monthly revenue

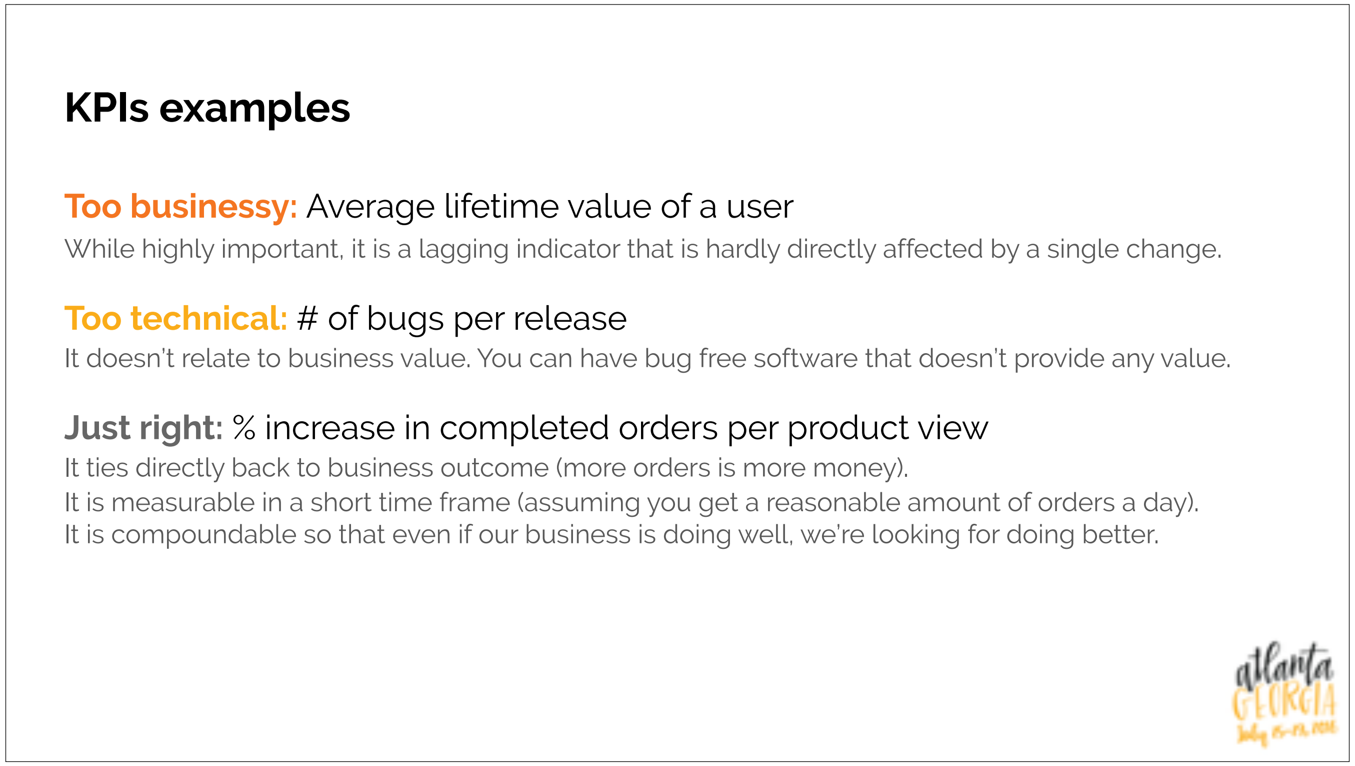
Too technical: # questions asked to waiters

Just right: Waiter labor hours saved

First one is a lagging indicator. Takes too long to change. Also is affected by way too many things.

The second one is easy to measure but doesn't really associate back to the business return.

The last one can clearly be associated to business impact and can easily be measured in a short timeframe.



KPIs examples

Too businessy: Average lifetime value of a user

While highly important, it is a lagging indicator that is hardly directly affected by a single change.

Too technical: # of bugs per release

It doesn't relate to business value. You can have bug free software that doesn't provide any value.

Just right: % increase in completed orders per product view

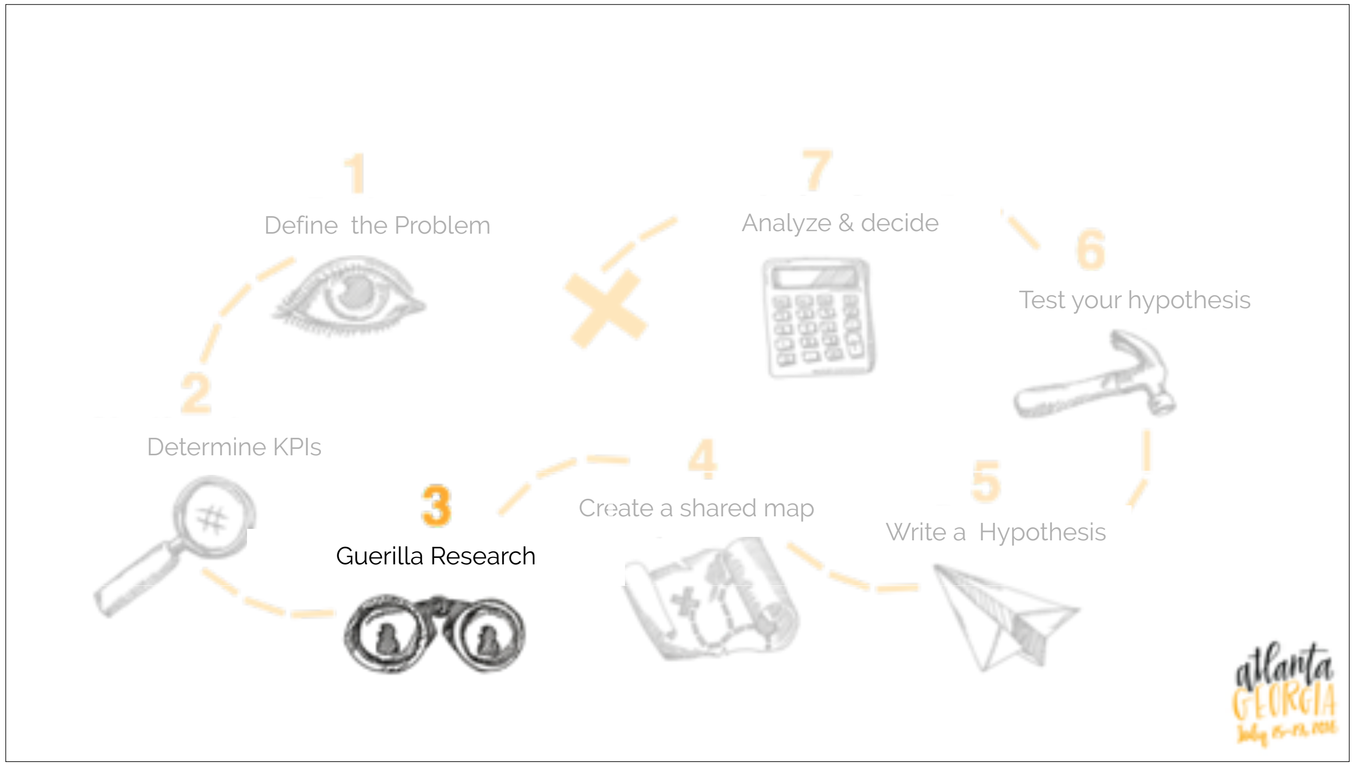
It ties directly back to business outcome (more orders is more money).
It is measurable in a short time frame (assuming you get a reasonable amount of orders a day).
It is compoundable so that even if our business is doing well, we're looking for doing better.

The first example here went too far toward the business team's desires. This metric is very important, but takes too long to collect (lagging indicator) and isn't actionable because many things could affect it.

The second example is too far on the technical side. Number of bugs can be collected easily and can be addressed right away, but aren't measuring what matters. Technical performance metrics have this same issue: their value is hard to quantify. What is the end result of increasing system throughput by 1%? Higher conversion rates for users? Lower hosted server costs? If one of those is the desired end state, it's better to start with that as a metric outright.

But the 3rd example can make both teams happy: it is directly related to business value, it's quickly measurable, and as a bonus, it can always get better.

It's important for the chosen metrics to have specific value to the company, and not be "vanity metrics" that make people feel good but don't actually mean the needle was moved in terms of value. For example, completing 100 user stories sounds great, but the things that got implemented might or might not have had concrete value for the company.



1 Define the Problem

2 Determine KPIs

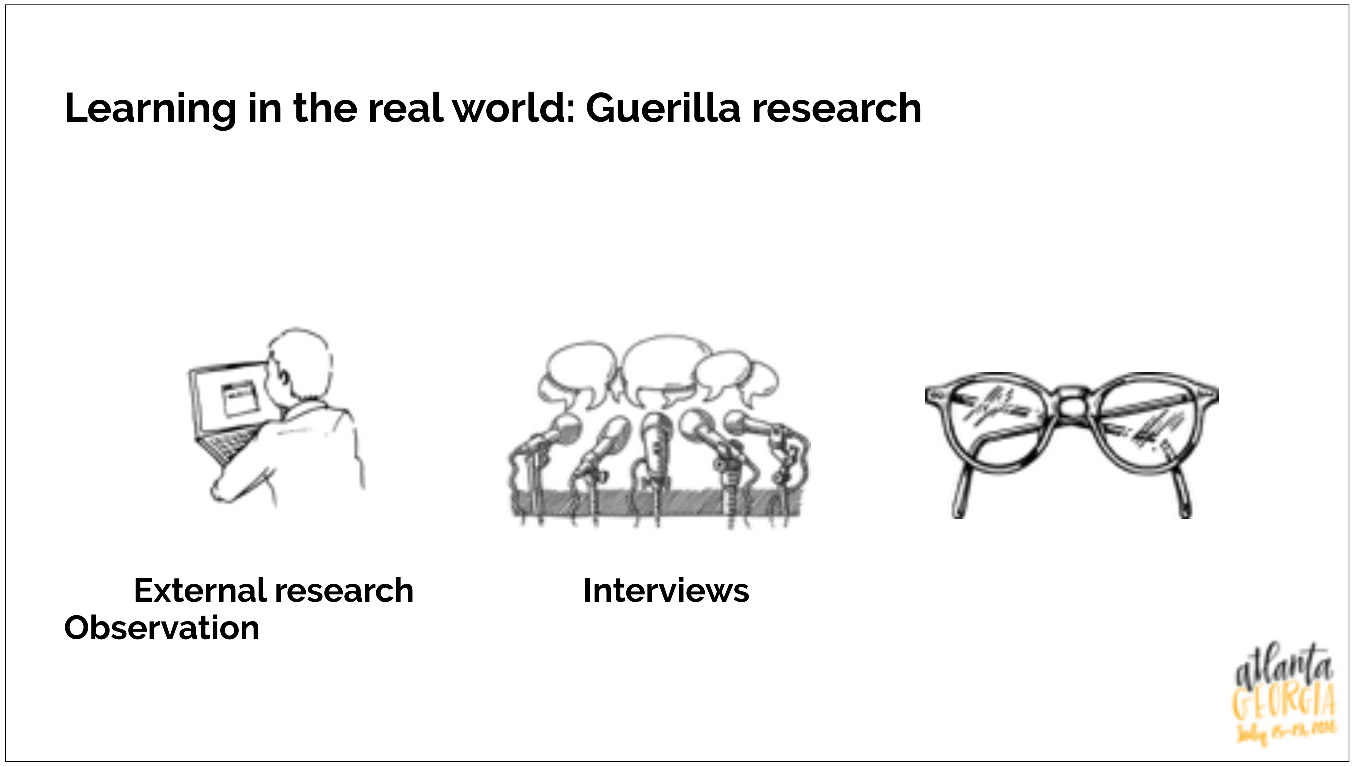
3 Guerilla Research

4 Create a shared map

5 Write a Hypothesis

6 Test your hypothesis

7 Analyze & decide



Learning in the real world: Guerilla research

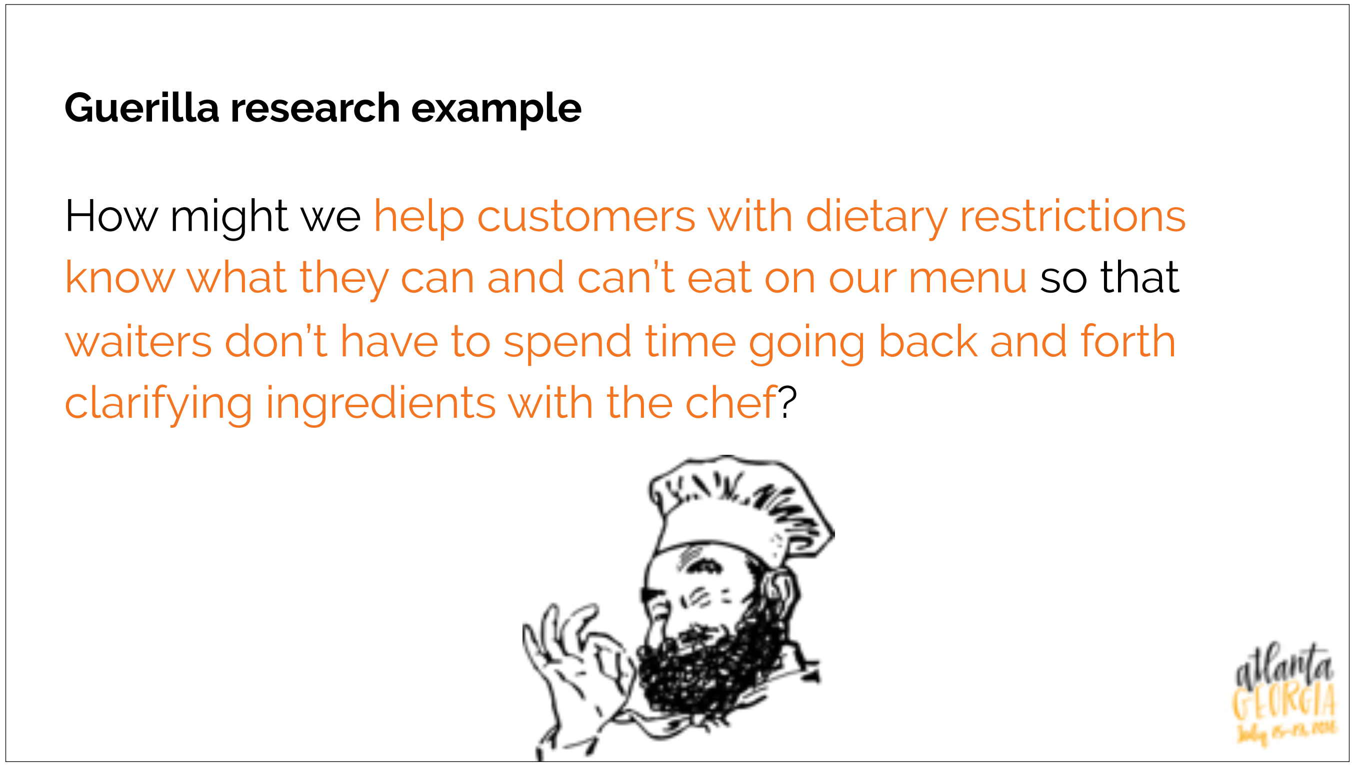
External research

Interviews

Observation

So you know the problem you're trying to solve and the metrics you're trying to affect. Before diving into an experiment, you'll need to know enough about the problem space to have a few hunches to test. You'll want to have a basic understanding of the highest opportunities within your vision, as well as the highest risks to its success. The best way to do this is through guerilla research, which has three main components (warning: there's an exercise coming up!):

1. External research: Other companies (inside or outside your industry) may have already researched this phenomenon. How can you leverage existing work to better understand the space? Think newspaper articles, industry reports, and blog posts. Or check out some actual solutions the competition has come up with to get a feel for what might be known by others -- but a word to the wise: they might have built their solutions without validating so take your competitive intelligence findings with a grain of salt.
2. Quick interviews: Get out of the building and head to a coffee shop, a park, a mall, or elsewhere to ask people open-ended questions about their habits, as they relate to your vision / problem space. You need fewer people and resources than you think - a couple of $5 coffee cards and some people who resemble your (assumed) target demographic. We're aiming for a heat map, not statistically significant data - 3-5 people can give us huge insight, and after that it tends to be diminishing returns for what you learn.
3. Observation / empathy for the user: If possible, observe users in action or even try out their jobs yourself. Designing for baristas? Trying being a barista for a day. Want to see the pain points in your app? Watch people use it. Here, it doesn't even need to be target users - it can even be a coworker or a relative for very early high level findings. This is important because people often DO things differently thatn they SAY they do them.

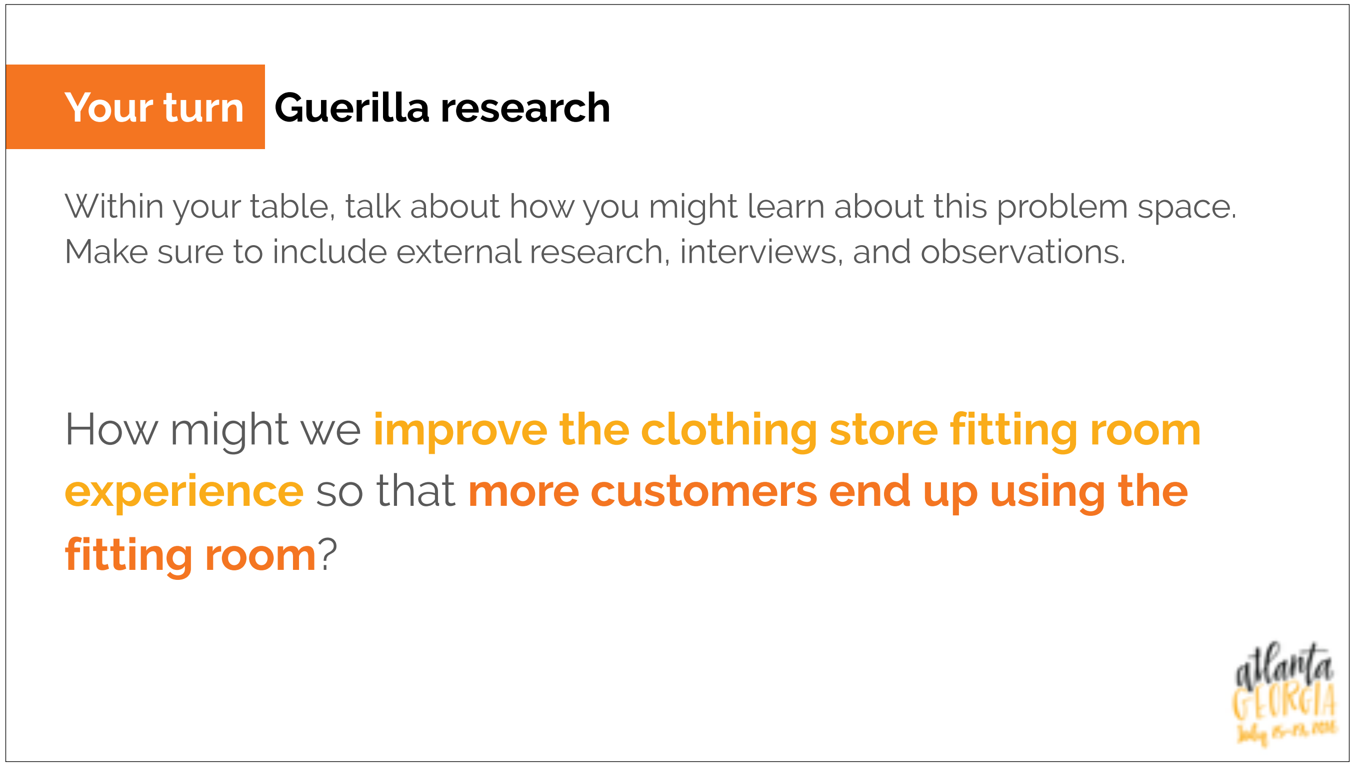


Guerilla research example

How might we help customers with dietary restrictions know what they can and can't eat on our menu so that waiters don't have to spend time going back and forth clarifying ingredients with the chef?

So if this was our problem space and problem statement, what should we plan on for guerilla research?

For External Research, I'd probably look at _________ and ______________, and maybe try a couple competitors' products. For Interviews, I'd talk to a few __________s about their past experiences with _________. And for Observation, I'd watch a few people use the existing tool and see what I could learn from watching similar people in _________.

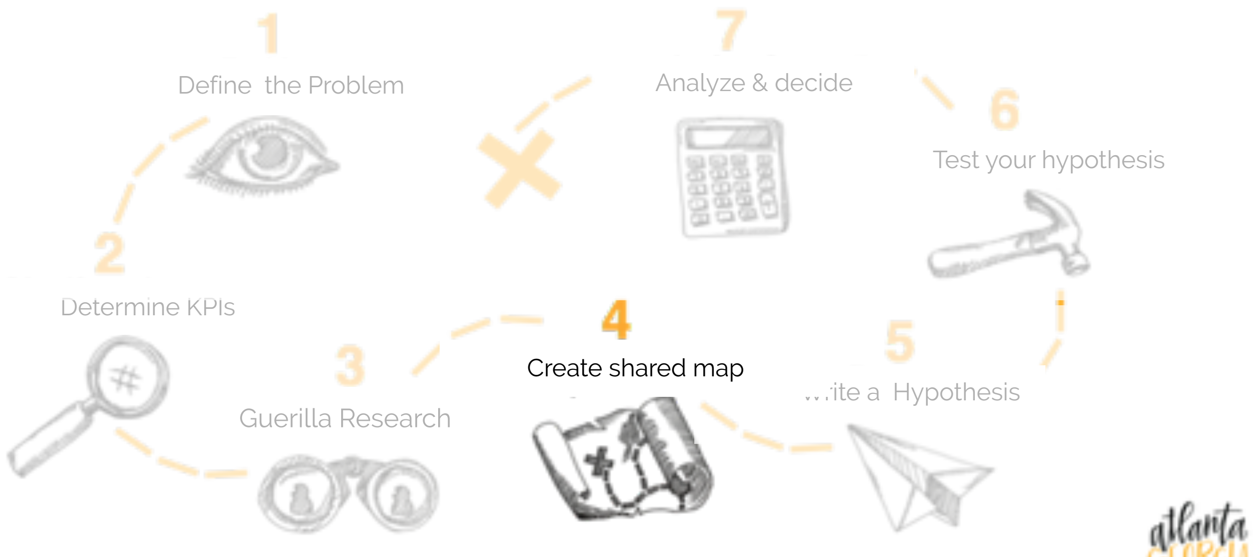


Your turn Guerilla research

Within your table, talk about how you might learn about this problem space. Make sure to include external research, interviews, and observations.

How might we **improve the clothing store fitting room experience** so that **more customers end up using the fitting room**?

Given the problem statement shown here, discuss as a table what your high level plan for guerilla research would be. Where would you first look for external research? Where you would go to interview people about the problem space? Who you might observe, and what would they be doing in what context?



1 Define the Problem

2 Determine KPIs

3 Guerilla Research

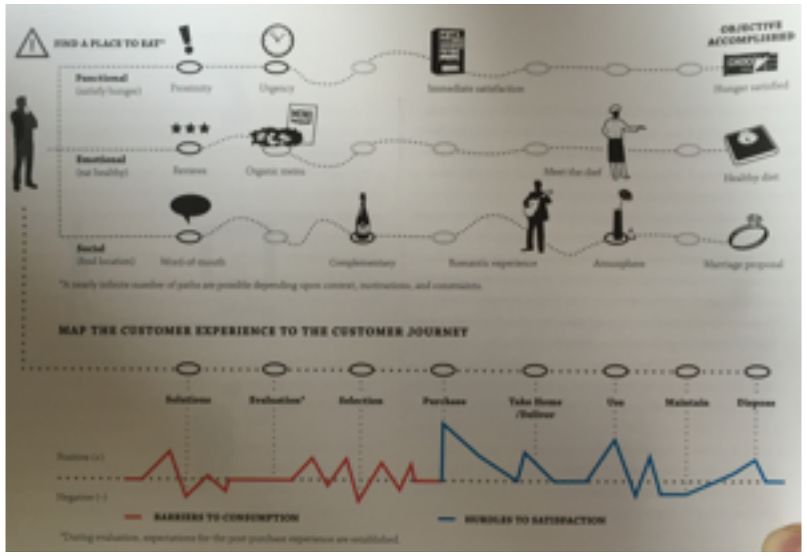
4 Create shared map

5 …ite a Hypothesis

6 Test your hypothesis

7 Analyze & decide

Create a shared experience map for understanding with high value focus areas



Source:
The art of opportunity - Mark Sniukas, Parker Lee, Matt Morasky

atlanta GEORGIA July 15-17, 2016

The goal of the shared map is to look at what your customers are currently experiencing in their current state. Then based on your guerilla research you can flag areas where your users are confused, frustrated, and overwhelmed, and what parts of the journey are currently working really well.

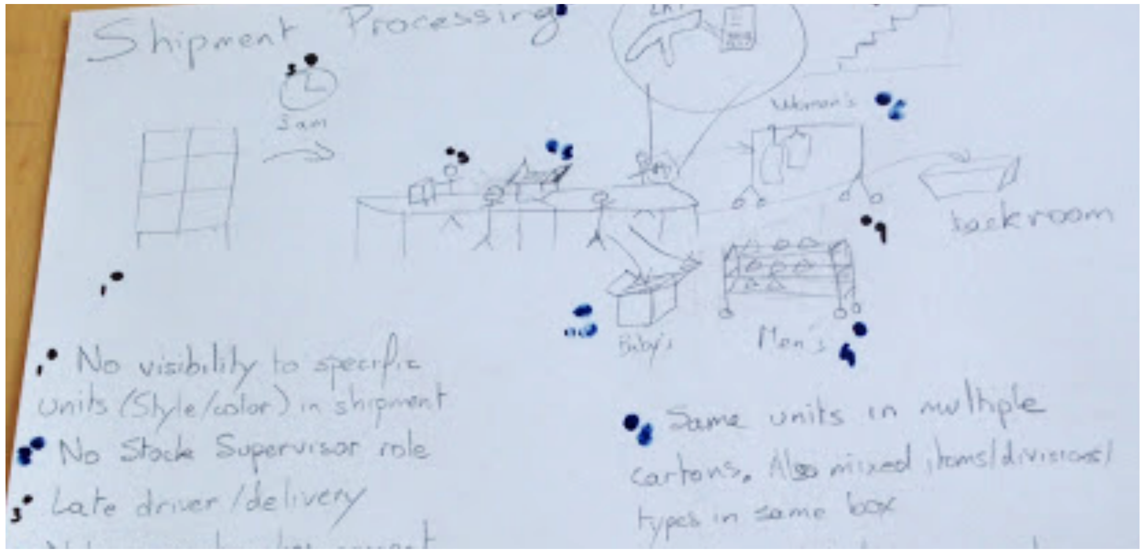
Typical business segmentation criteria includes things like age, gender, income, etc., but those aren't enough to show you where there are opportunities for improvement, nor do they help your team empathize with those you're trying to serve.

1) Draw the user journey

2) Overlay the pain points and opportunities you discovered from

Explain what pain points are

Share that this can take a lot of different forms.

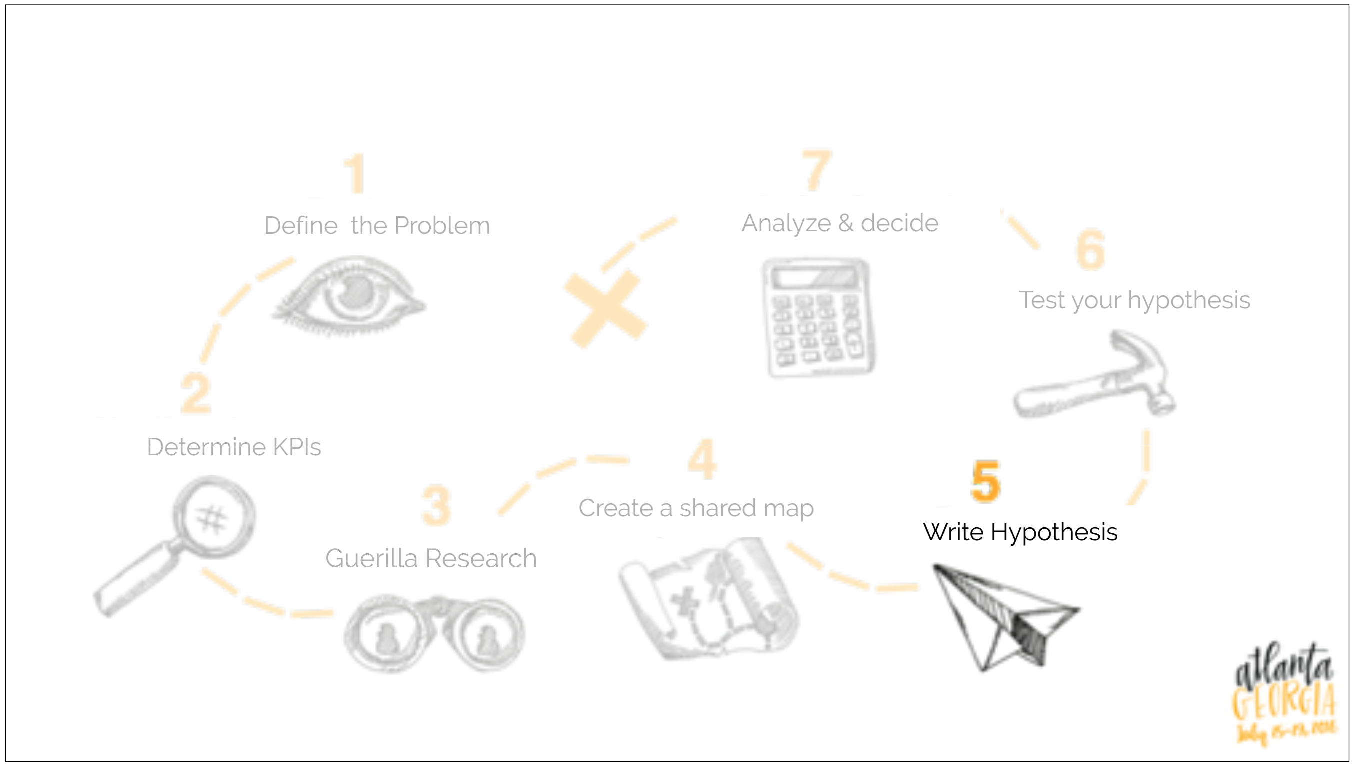


Shipment Processing

- No visibility to specific units (Style/color) in shipment
- No Stock Supervisor role
- Late driver/delivery
- Same units in multiple cartons. Also mixed items/divisions/types in same box

Explain what pain points are

Share that this can take a lot of different forms.



1 Define the Problem

2 Determine KPIs

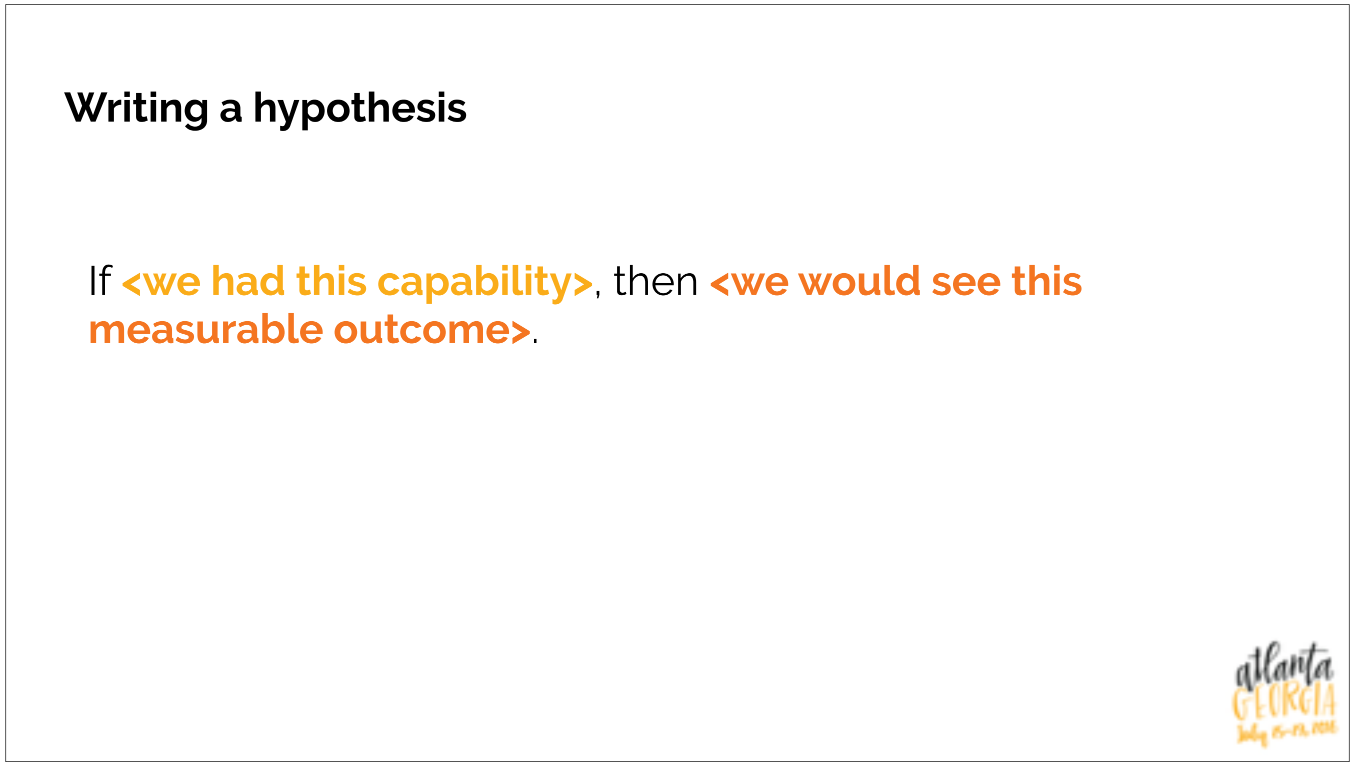
3 Guerilla Research

4 Create a shared map

5 Write Hypothesis

6 Test your hypothesis

7 Analyze & decide

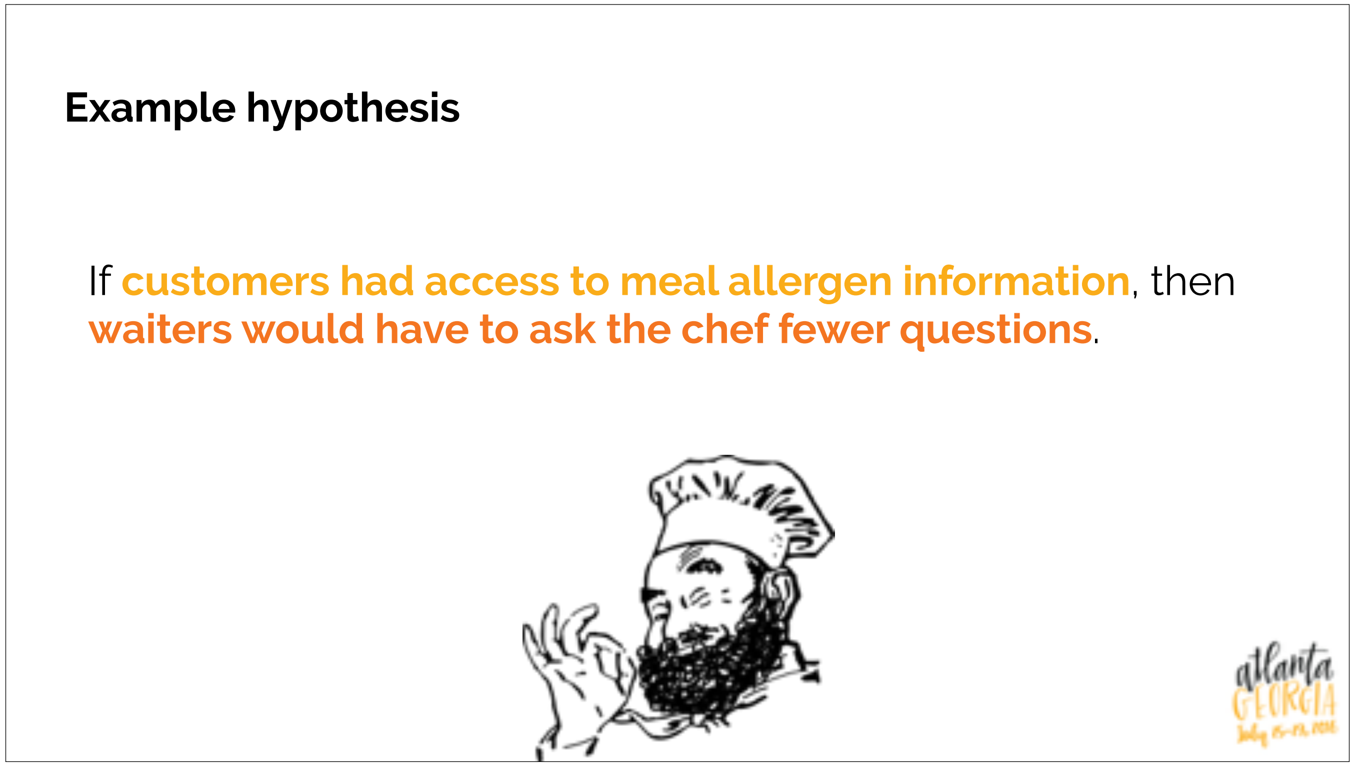


Writing a hypothesis

If **<we had this capability>**, then **<we would see this measurable outcome>**.

atlanta GEORGIA

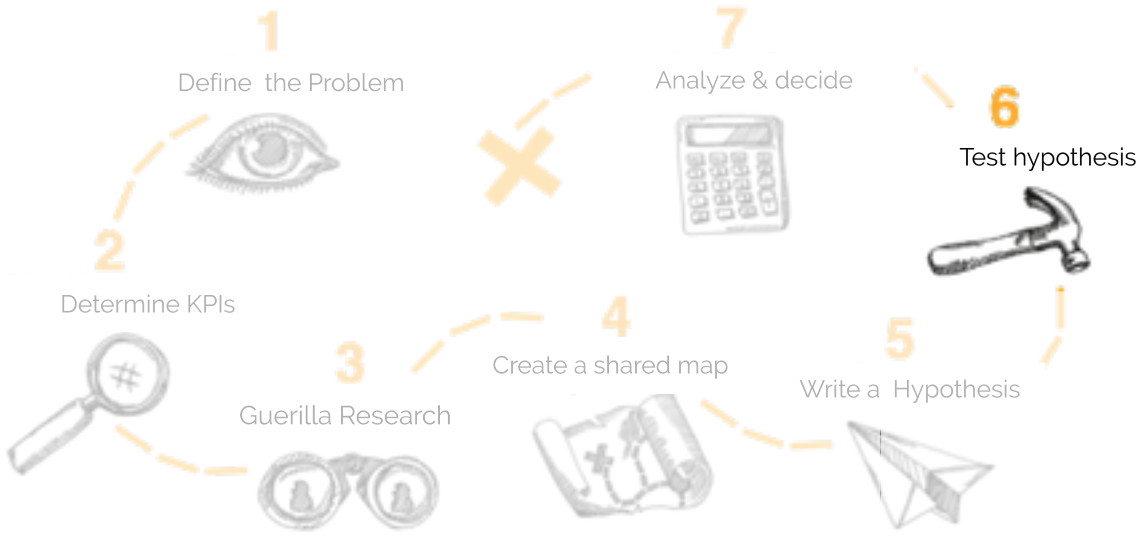
The shared map gave you prioritized opportunities to start learning about first. Here’s a great way to understand if your assumptions about the top opportunity are correct. At its core, the hypothesis is a simple tool to help put into words how you believe your idea will impact the outcome.



Example hypothesis

If **customers had access to meal allergen information**, then **waiters would have to ask the chef fewer questions**.

Here’s an example with the restaurant.



1 Define the Problem

2 Determine KPIs

3 Guerilla Research

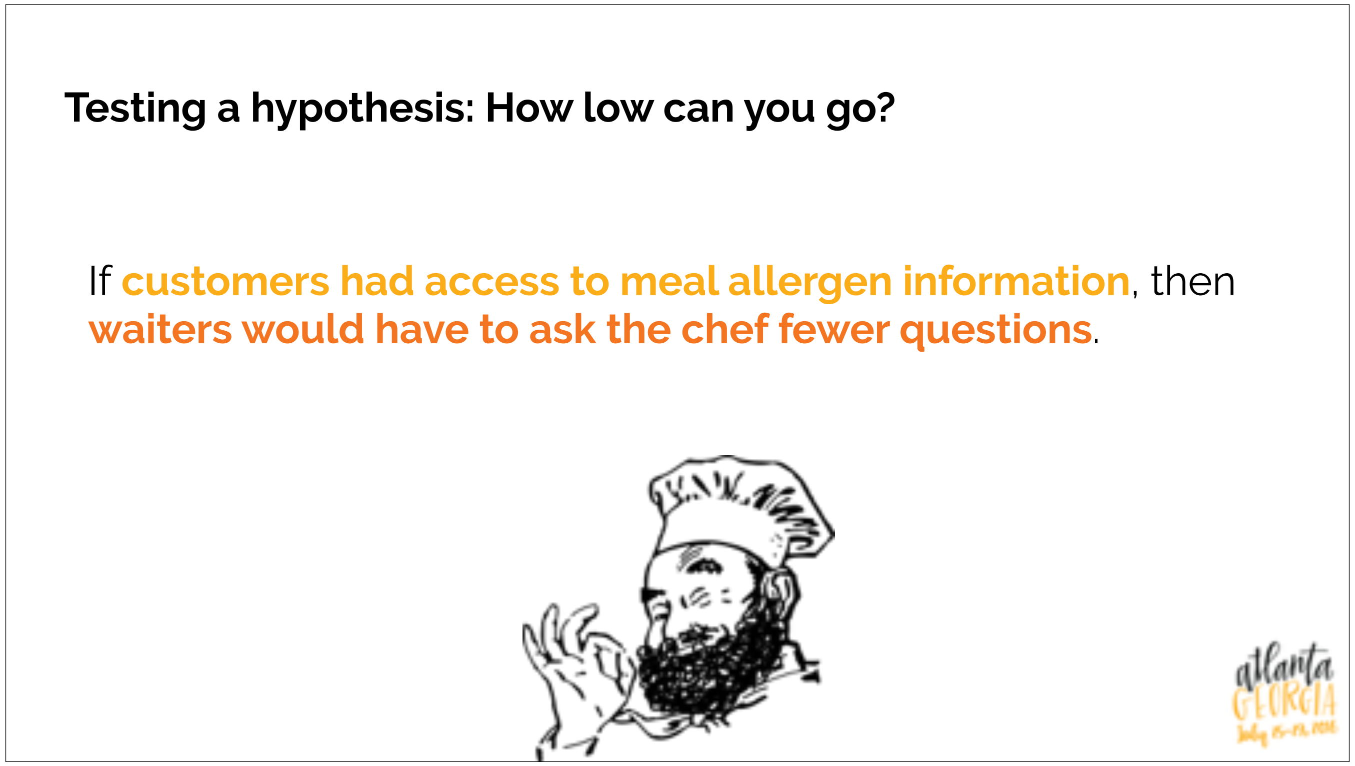
4 Create a shared map

5 Write a Hypothesis

6 Test hypothesis

7 Analyze & decide

atlanta
GEORGIA
July 15-17, 2016



Testing a hypothesis: How low can you go?

If **customers had access to meal allergen information**, then **waiters would have to ask the chef fewer questions**.

How could you look at this hypothesis and learn as quickly as possible whether or not it is true? You'll do this after us, so pay attention…

1. Make an app for customers where they can log in and tell us what they're allergic to and then we can filter the menu based on what they can have… BUT Why would you spend so much time building that if you don't even know for sure that people would find that
2. Maybe have a big TV where cooks can enter the latest ingredient information daily and customers can see it as they walk in? BUT is there a way to do it without needing to buy a TV and the cooks not need to update it daily?
3. What if we reprint the menus with all allergen information? BUT is there a way to show easy to understand allergen information without printing menus?
4. How about a bunch of GF, V, etc. stickers on the current menus?

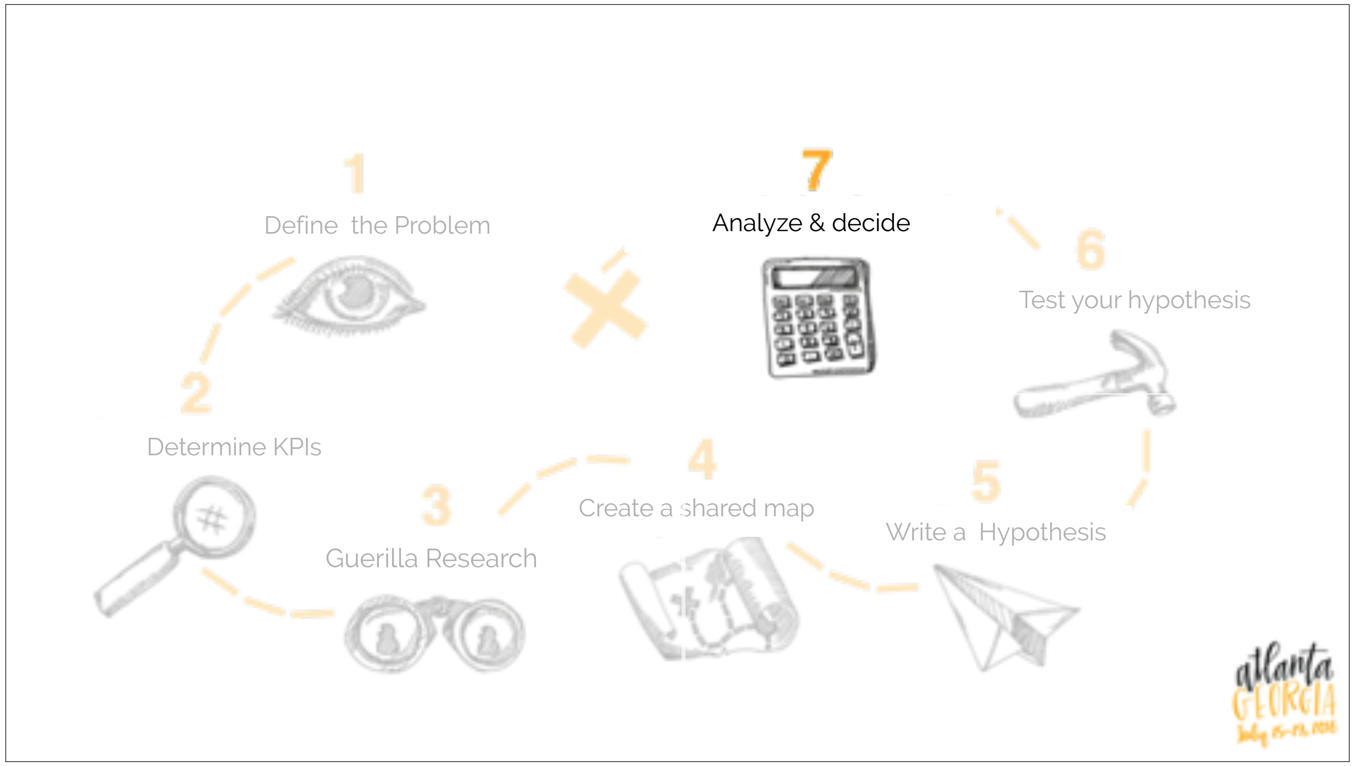
It might turn out that you need stickers, or it might mean that you learn that you do need some features in an ap, but the point is that you learned that without spending the time and money of actually building it before you know for sure.

Your turn How low can you go?

If **fitting rooms provided entertainment,** then **more customers would use the fitting room**.

With a partner, take turns coming up with ways you could test this, each test idea being cheaper and faster to implement than the last. Come up with a total of 4 test ideas.

atlanta
GEORGIA



1 Define the Problem

2 Determine KPIs

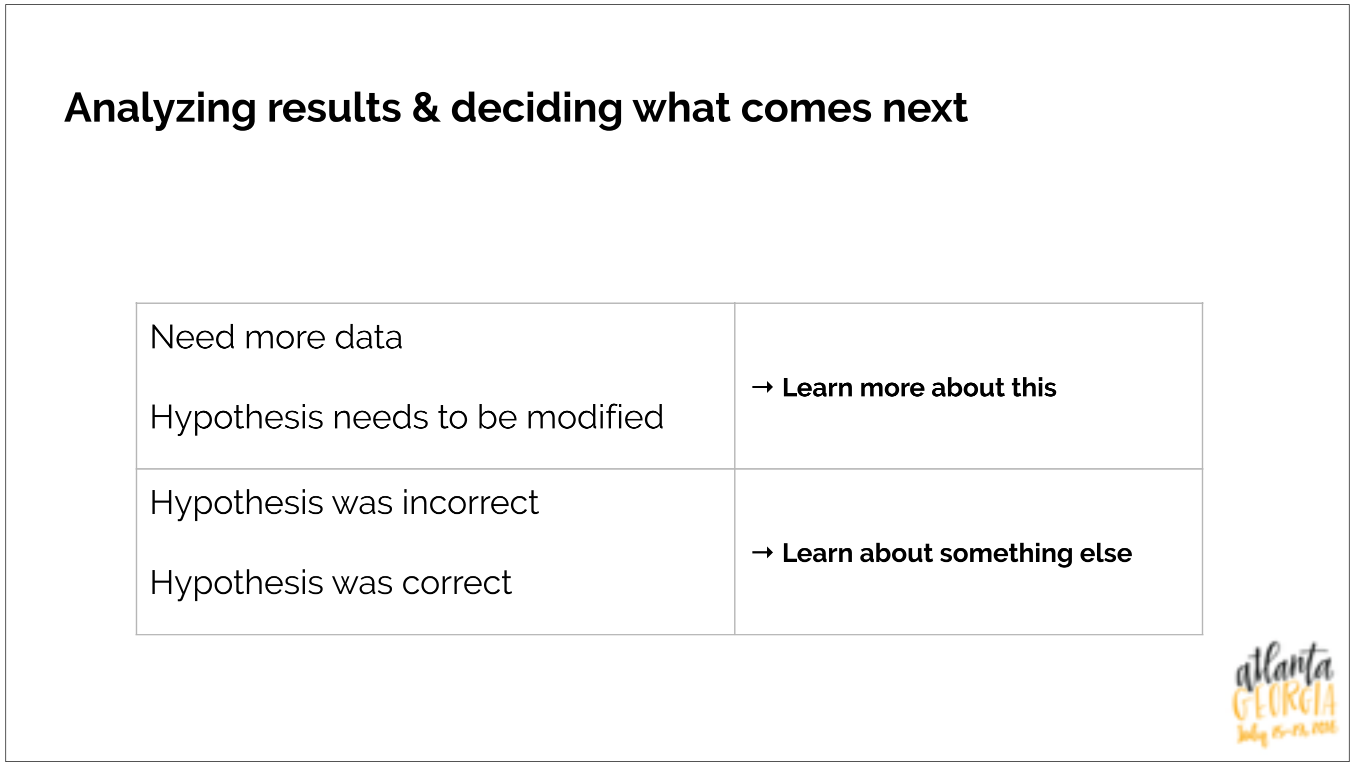
3 Guerilla Research

4 Create a shared map

5 Write a Hypothesis

6 Test your hypothesis

7 Analyze & decide



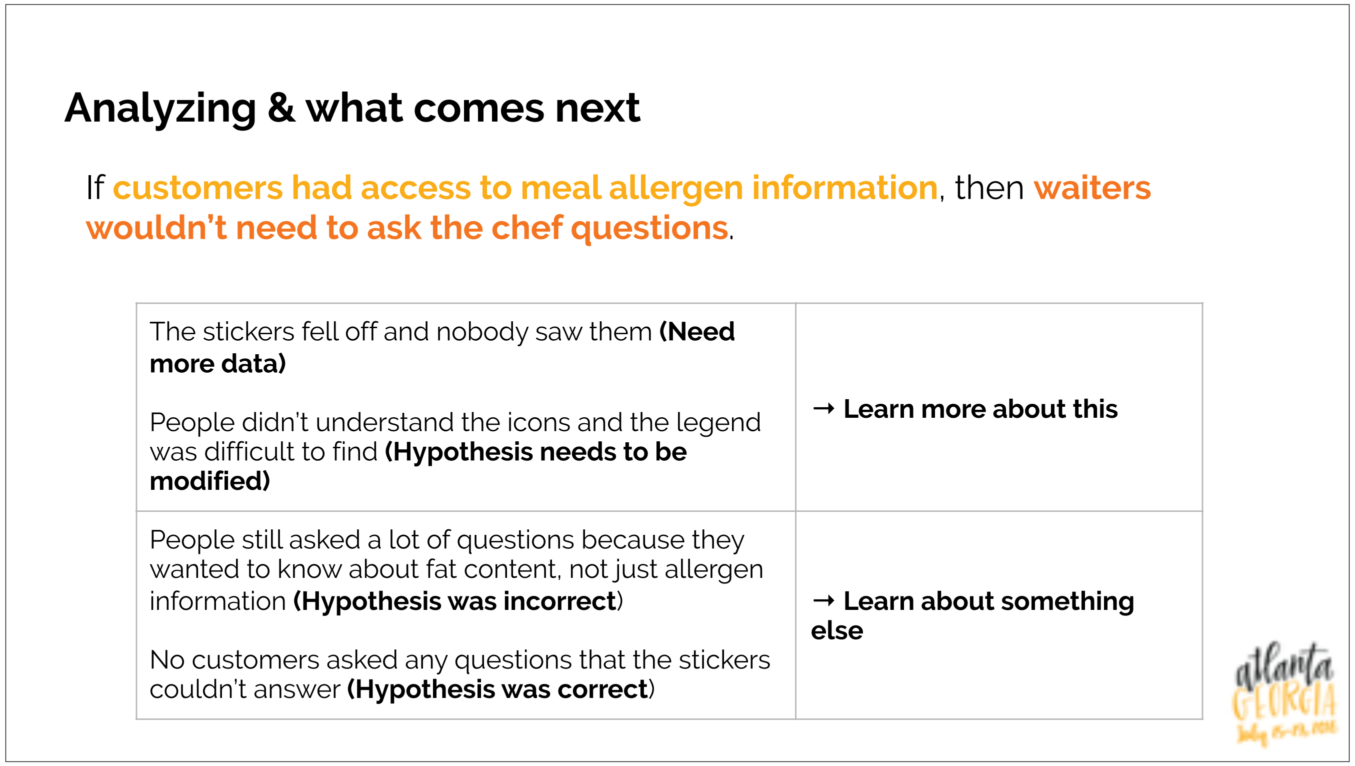
Analyzing results & deciding what comes next

Need more data Hypothesis needs to be modified	→ **Learn more about this**
Hypothesis was incorrect Hypothesis was correct	→ **Learn about something else**

Two possible outcomes: learn about it some more. Learn about something else.

On the first scenario, we might have had a problem with our experiment and not collected the right data or just needing higher fidelity information to keep going. Or, we might have looked at something that wasn't quite the right thing but we feel like there is still value in there and we just need to change our hypothesis a bit.

On the second scenario, we might have found out that our hypothesis was incorrect. Given the capability we provided, we did not see the outcome we wanted and that is great! It means we spend very little to discover our idea was not going to be successful. That's a great cost saving for the organization!
On the other hand, we might discover that we are correct and that hypothesis of ours yields the results we expected and we can head towards building it or just explore another one.

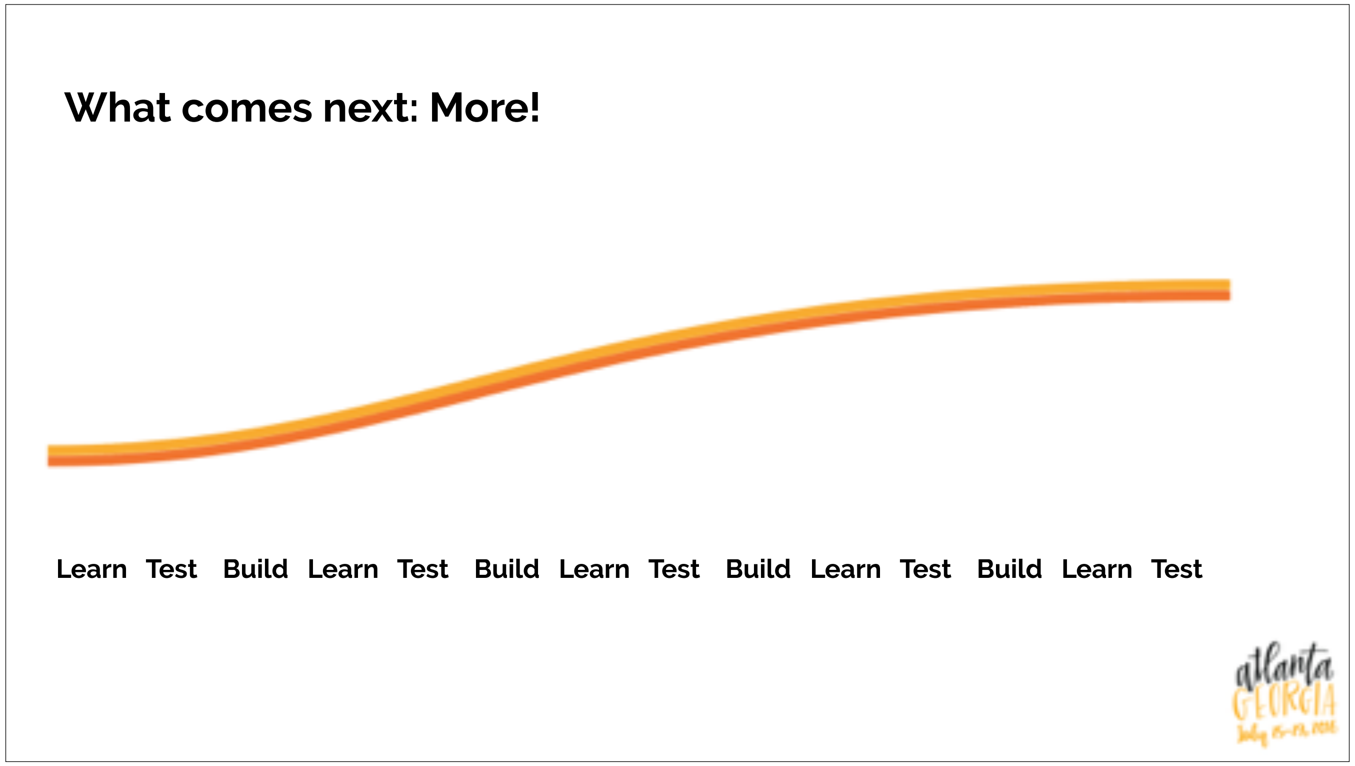


Analyzing & what comes next

If **customers had access to meal allergen information**, then **waiters wouldn’t need to ask the chef questions**.

The stickers fell off and nobody saw them **(Need more data)** People didn’t understand the icons and the legend was difficult to find **(Hypothesis needs to be modified)**	**→ Learn more about this**
People still asked a lot of questions because they wanted to know about fat content, not just allergen information **(Hypothesis was incorrect**) No customers asked any questions that the stickers couldn’t answer **(Hypothesis was correct**)	**→ Learn about something else**

For our restaurant example, here are a few examples of each potential outcome.



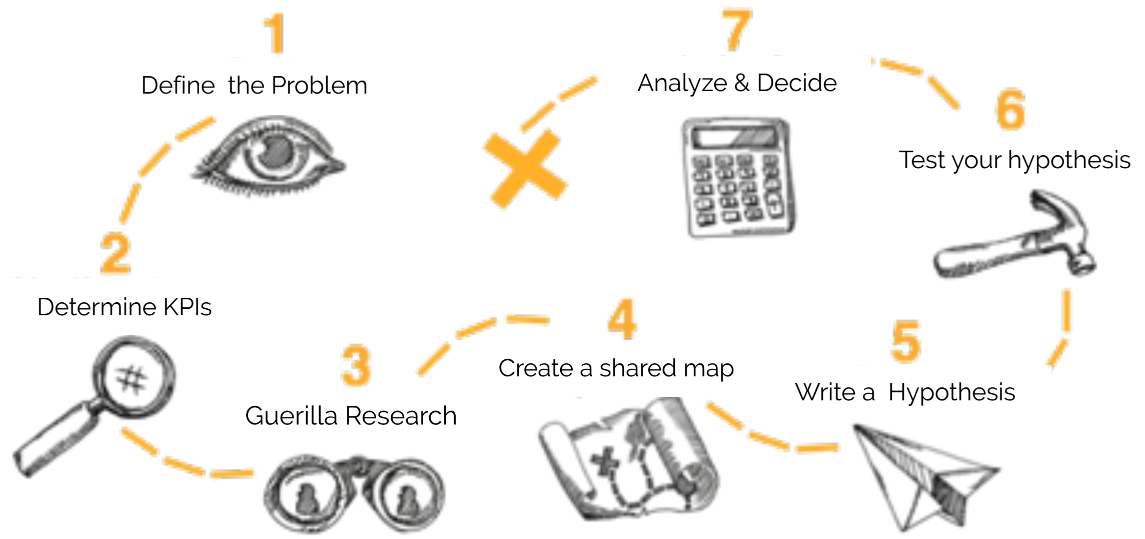
What comes next: More!

Learn Test Build Learn Test Build Learn Test Build Learn Test Build Learn Test

atlanta
GEORGIA

This is not a “ta-da!” at the end of the project; this is a continuous cycle throughout the life. Low fidelity tests early on have fewer data points and need less analysis; later, higher fidelity experiments will have more data points and thus need more analysis

Lean Experimentation



1 Define the Problem

2 Determine KPIs

3 Guerilla Research

4 Create a shared map

5 Write a Hypothesis

6 Test your hypothesis

7 Analyze & Decide

Questions & Discussion

Thank you!