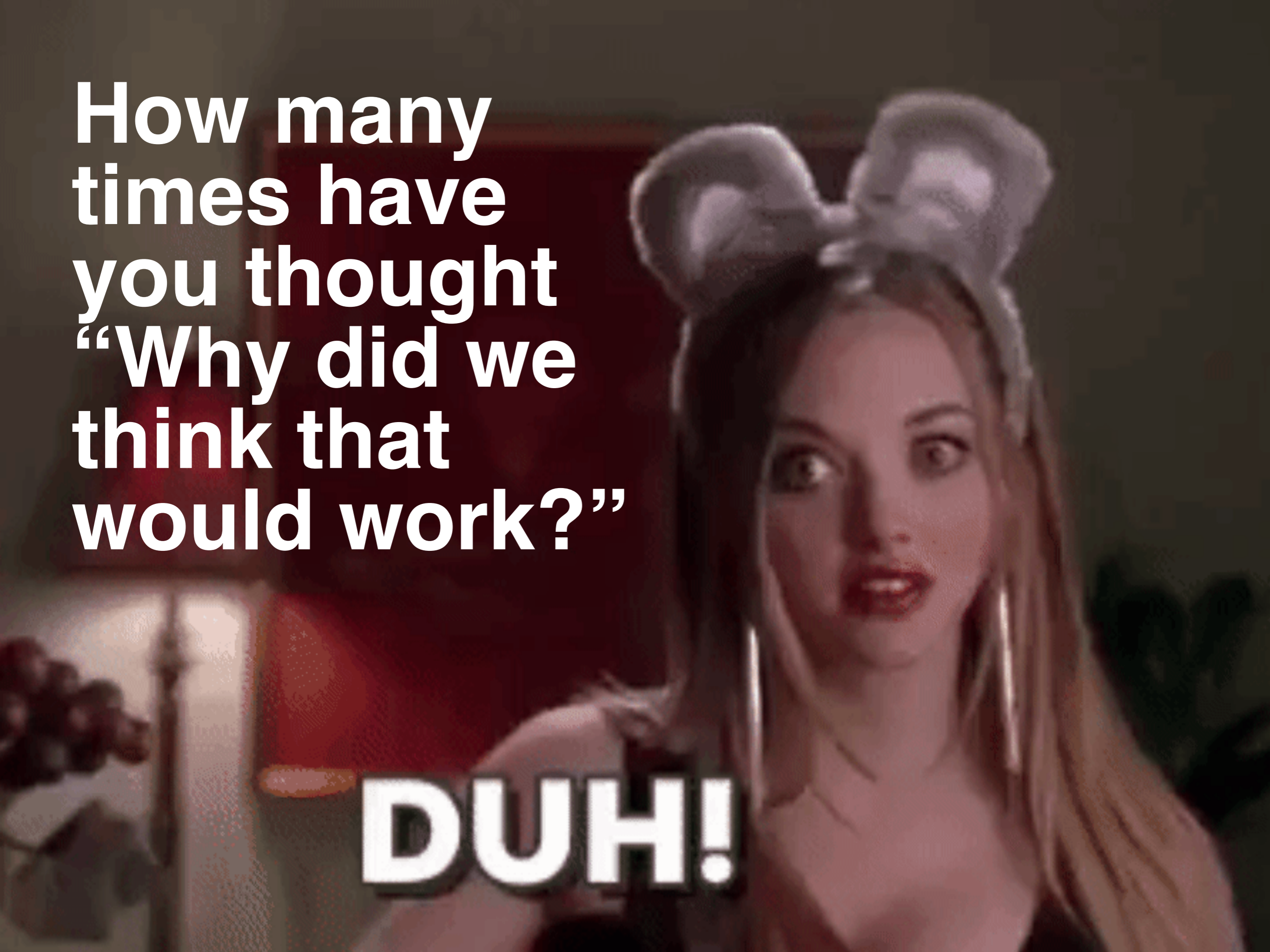


**How many times
has your team
had to rebuild
something?***

*Great idea— it just didn't make sense in practice

**How many
times have
you thought
“Why did we
think that
would work?”**

DUH!



Testing can help!

User Testing Walk Through
Natalie Levy-Costa



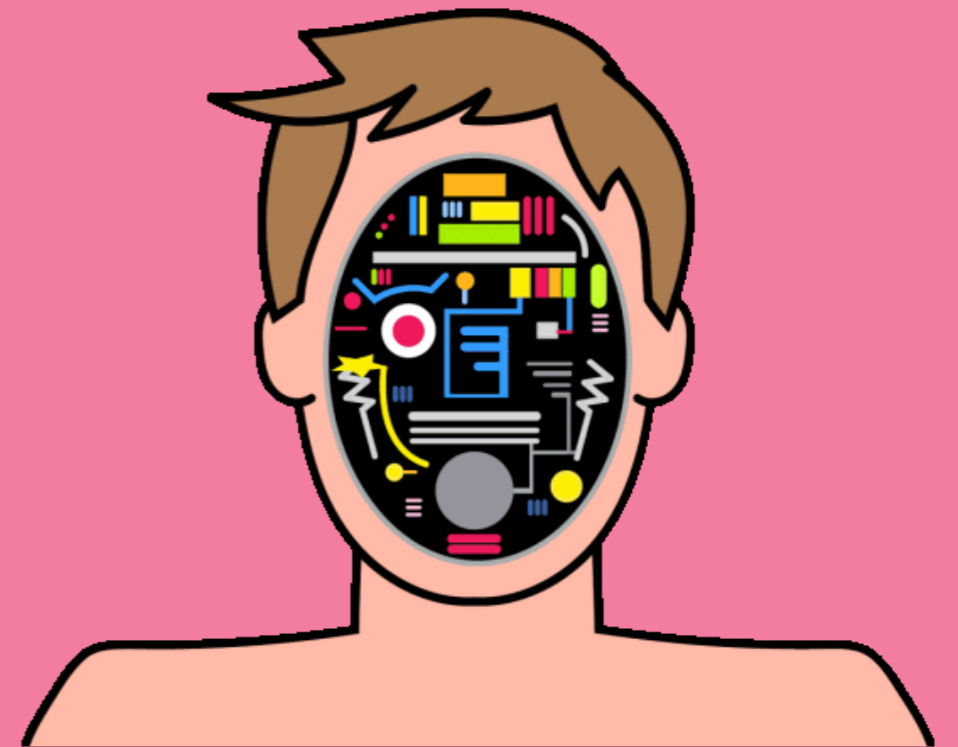
But what is this thing called “user testing”?

Exactly what it sounds like: introducing sample users to a new feature or product to vet viability before we spend \$\$\$ building it.



Why do we test?

Because no matter how much I wish they were, people are not predictable robots.



What is the advantage?

In situations where mistakes can get very expensive, user testing is a quick and cheap way to:

- **Validate** a hypothesis in a low-stakes environment
- **Iterate** behind-the-scenes without frustrating or confusing customers as the system evolves
- **Fail fast** without losing momentum

What happens when we fail to test?

“This is going to be the fourth time I have been asked to update this information. Why can't you get things fixed so that the information stays in place? In case your wondering, I'm not very happy right now.”

(Real feedback from a real user)

**“Okay,
Natalie, but
why should
I care?”**

Why should testing matter to you?



Fixing an error after development is up to 100 times more expensive than it would have been before. And an estimated 50% of engineering time is spent doing rework that could have been avoided.

Fine, I get it. Let's get down to the nitty-gritty....

Types of Design Testing:

- **Qualitative**

- Usability testing*

- Most often referred to as “user testing”

- Moderated or unmoderated

- Usability (5 users), Preference (20-30 users),
or Foundational Discovery (varies)

- In-person focus groups

- **Quantitative**


- In-market studies

- A/B testing

*This is what we will spend most of
our time talking about today

Testing for

Usability

- **Why 5 to 8 users?**
 - After testing with between 5 and 8 users, you begin to reach a point of diminishing returns
- **What can we learn?**
 - Does this design or interaction make sense?
- **When is this the right choice?**
 - Rule of Thumb: **Always** user test! 
 - When we're moving as fast as possible
 - When you are testing one key change, something brand new to your users, or plan for multiple tests



**Five users
will most
likely
detect
about 85%
of the
problems
in an
interface**

Testing for

Preferences

- **Why 20 to 30 users?**
 - At this point, you reach a 90% confidence interval for insights into preferences for **different user types**
 - This number can span multiple smaller/iterative tests—see slide 11
- **What can we learn?**
 - Behavioral insight into the proposed solution
- **When is this the right choice?**
 - You are trying to solve for an “unknown”, have a little data but need validation before going to market, are improving a previously tested MVP that is “almost there”
 - You have the runway to do so (time & urgency) 🕒

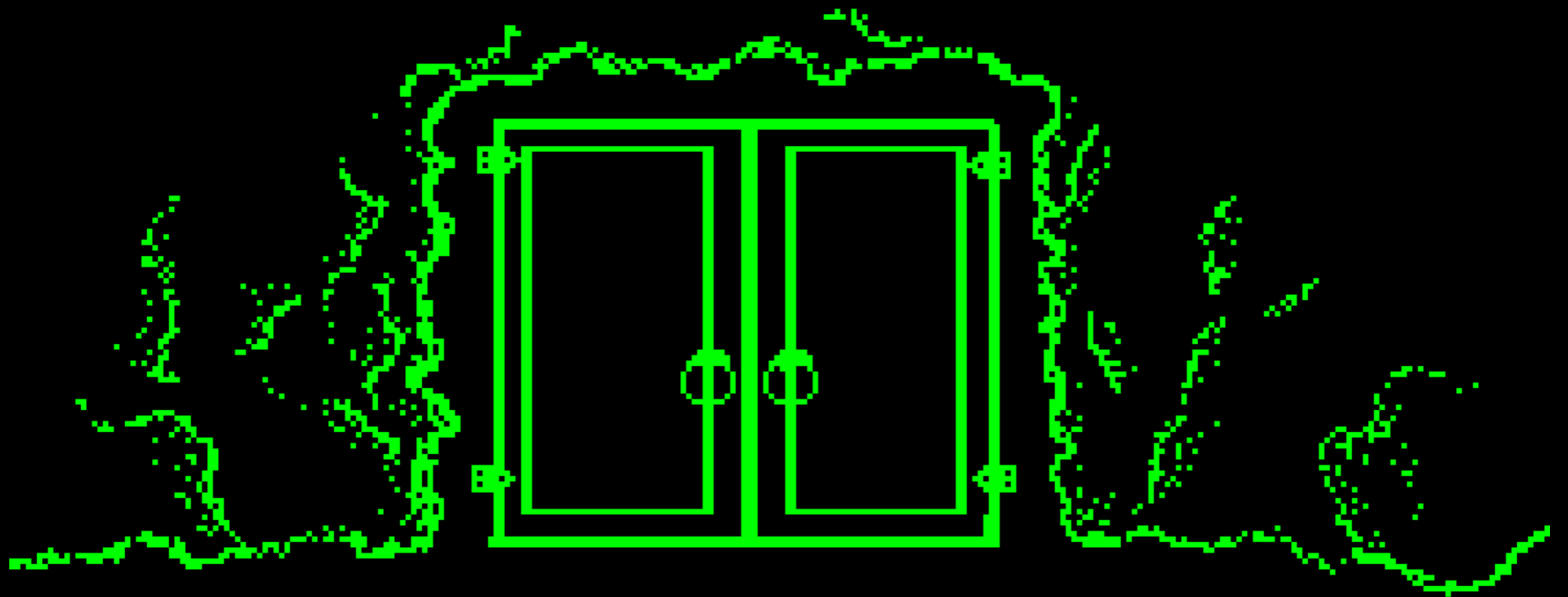


Testing can ensure that products aren't just usable, but also enjoyable across segments

Testing for

Foundational Discovery

Discovery is where we venture into the unknown!



ENTER THE DUNGEON?

> hell yeah

(Is this the right dungeon?)

In-market and A/B testing

- **What can we learn?**
 - How users respond to changes to an existing system, user behavioral patterns
- **When is this the right choice?**
 - When we need to choose between several directions, when we want to change an existing product or user behavior, when you want to experiment in a live setting
 - When you know the data you need and have a means of capturing that data

So why not just jump straight to in-market testing?



Deciding factors here are **time** and **risk**:

- How much do we want to invest in an unknown entity?
- How sure are we that this will work?
- How sure are we that there is an appetite for this kind of product or change?
- How much data do we have?
- How much of an impact do we need to make?

*According to
FastCompany:*



Industry surveys show that every dollar invested in UX will bring \$2 – \$100 in return. Not only does testing reduce costs, but you'll also craft an experience that's more pleasing to your customers.

Testing best practices

- 1. Test as early as possible:** The earlier you test, the easier it is to make changes, and test again. The design process should be iterative and allow for a feedback loop of “Build. Measure. Learn. Repeat.”
- 2. Understand your objectives before you test:** Always be clear on your goals before the test begins. This way you fully understand what areas of feedback to look out for while the test is taking place. Sometimes while testing, unprompted feedback may be shared. While the unprompted feedback is important, always focus on your main objectives.
- 3. Carefully prepare questions and tasks:** Questions and tasks need to be clear to the user without leading. Any type of leading questions or task prompts could skew results. When creating your test plan, be sure to minimize the amount of questions and tasks per test so as not to overwhelm users.
- 4. Don't try to solve everything at once:** Tests should have specific objectives as to what they are trying to understand. If the problem or objective is too big, consider breaking down the test into two parts. **Iterate, Iterate, Iterate.**
- 5. Get the whole team engaged in the process:** Once you have your findings, share with the team! Using video clips and pull quotes from the tests can help the team understand the real problems that the users are facing.



Conclusion



- Different tests are best for different problems
- Testing prototypes is fast!
- Testing saves on development time and budget
- Testing leads to better user experiences

Questions

