

How to Program As a Pair



FULLSTACK

What This Lecture Covers



How pairing combines your minds into one super-mind.



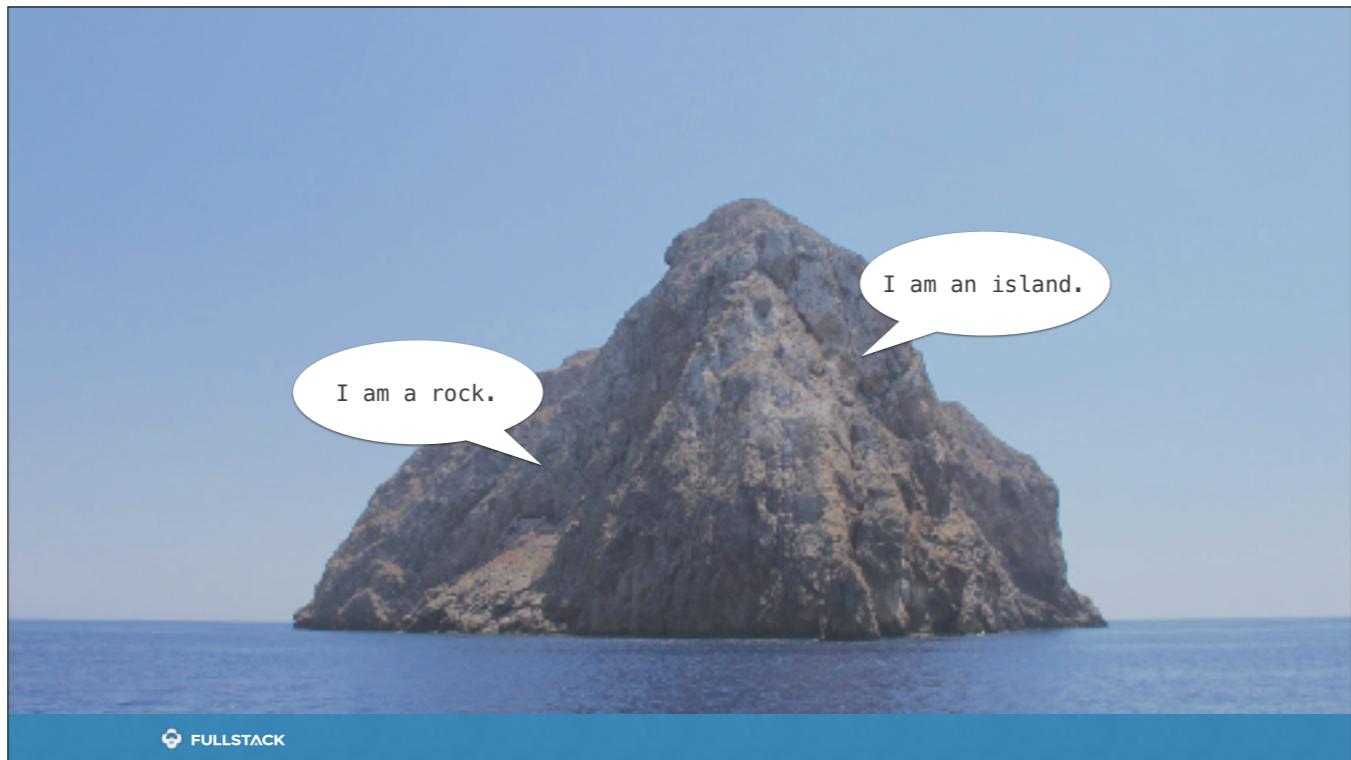
How to navigate

How to drive

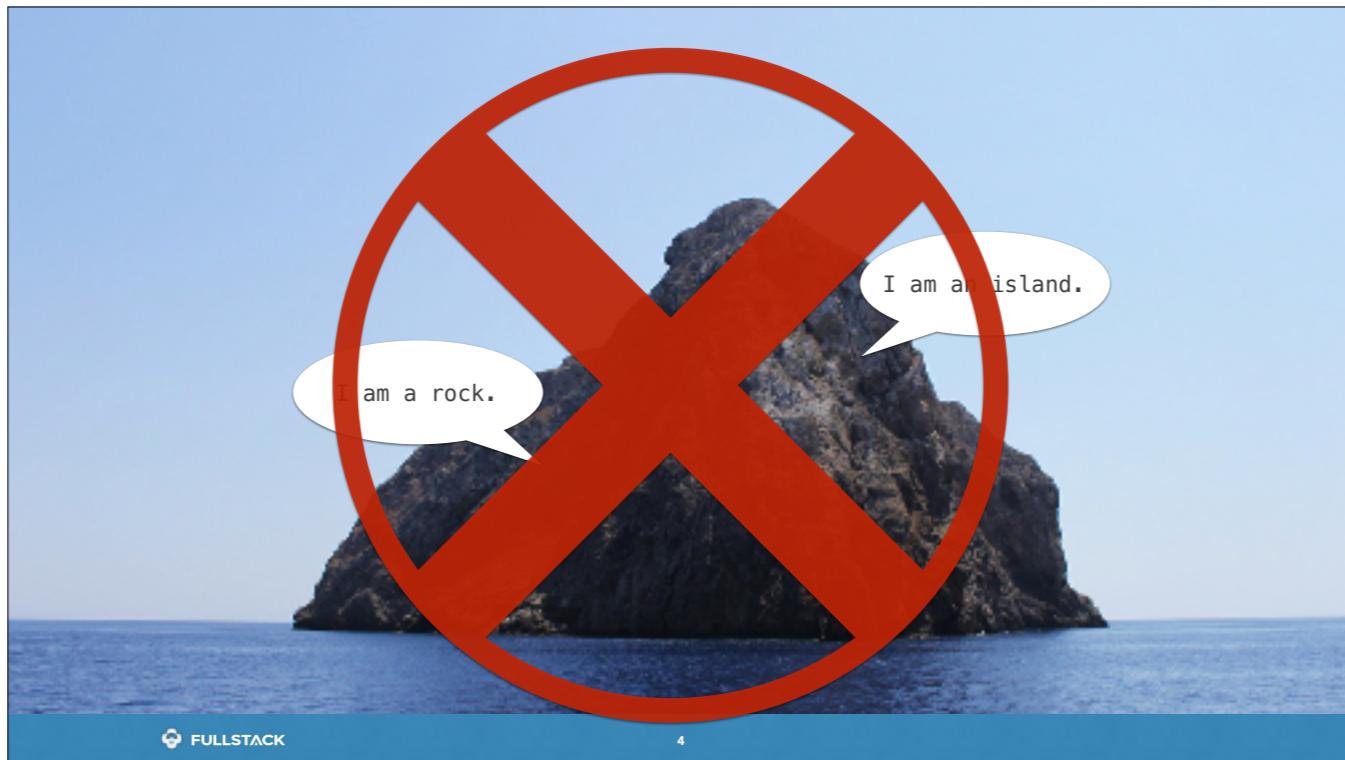


Specific instructions on how to work together effectively when things aren't going well.

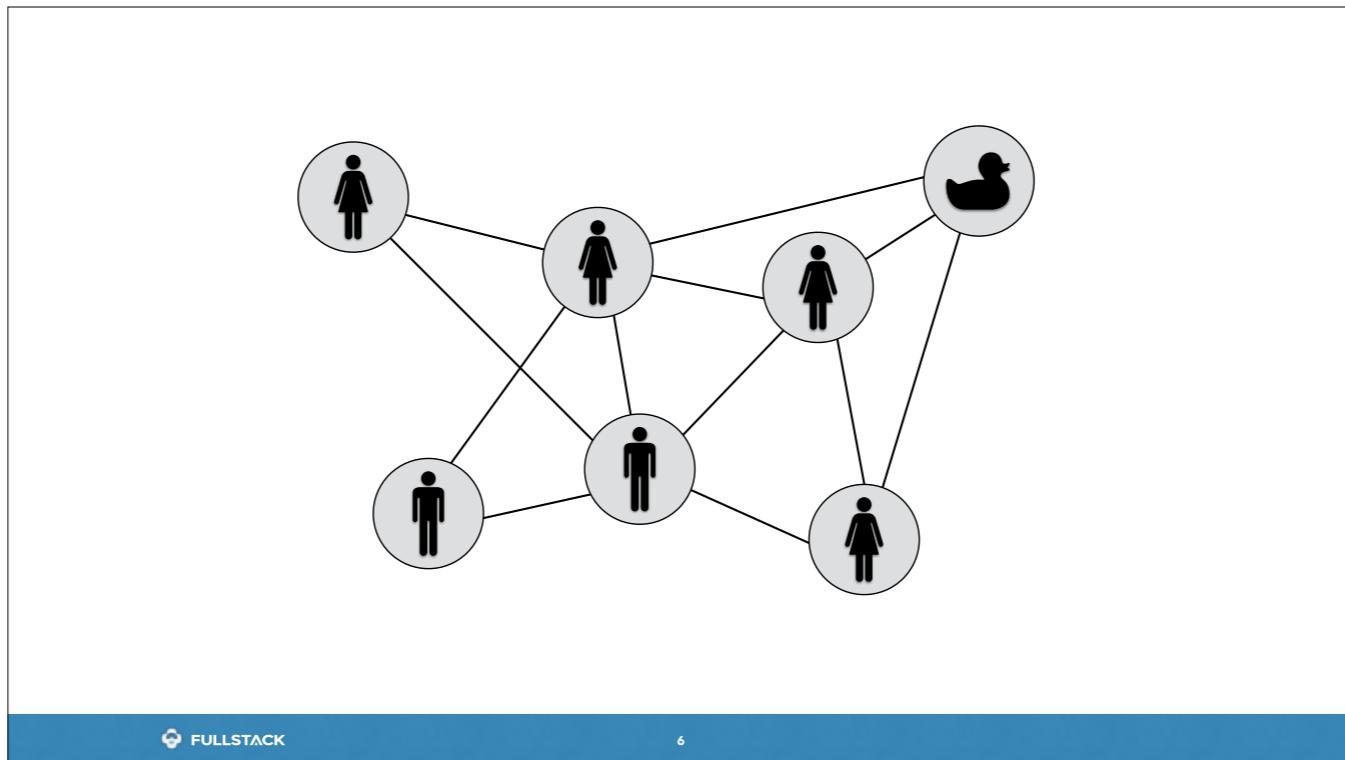




 FULLSTACK

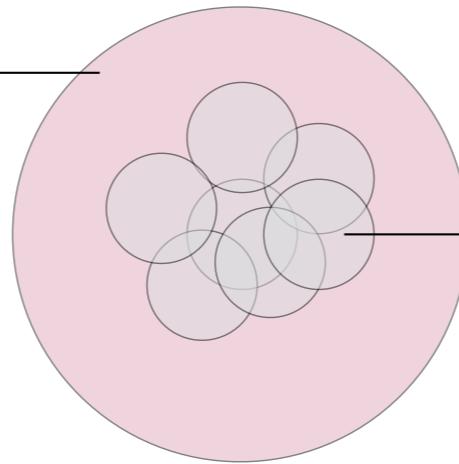


No programmer is an island.



We work in teams.

Everything that can be known/done.

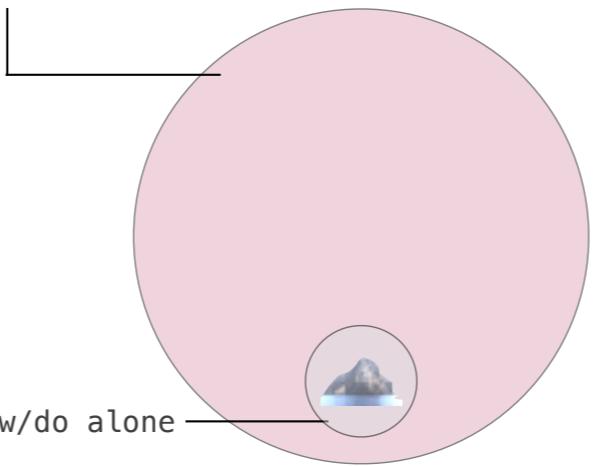


What each member of the team knows/can do.

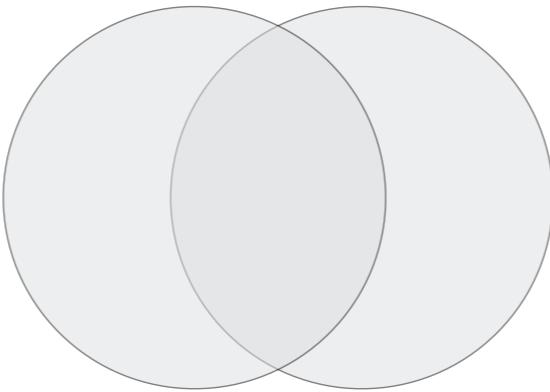
We work in teams to maximize our knowledge and capabilities.

Everything that can be known/done.

What you can know/do alone

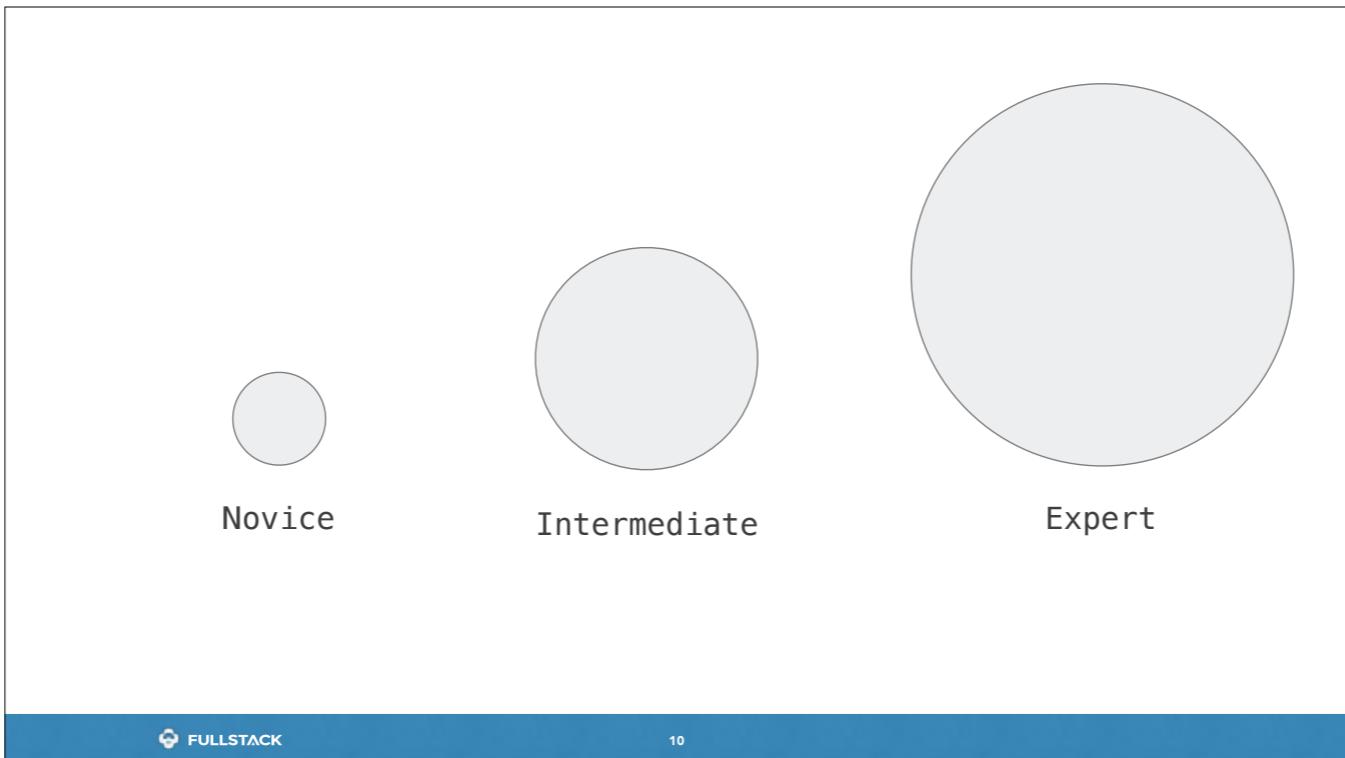


:(
Sad, lonely islands would be smarter, taller, and be better looking if they were members of a team.



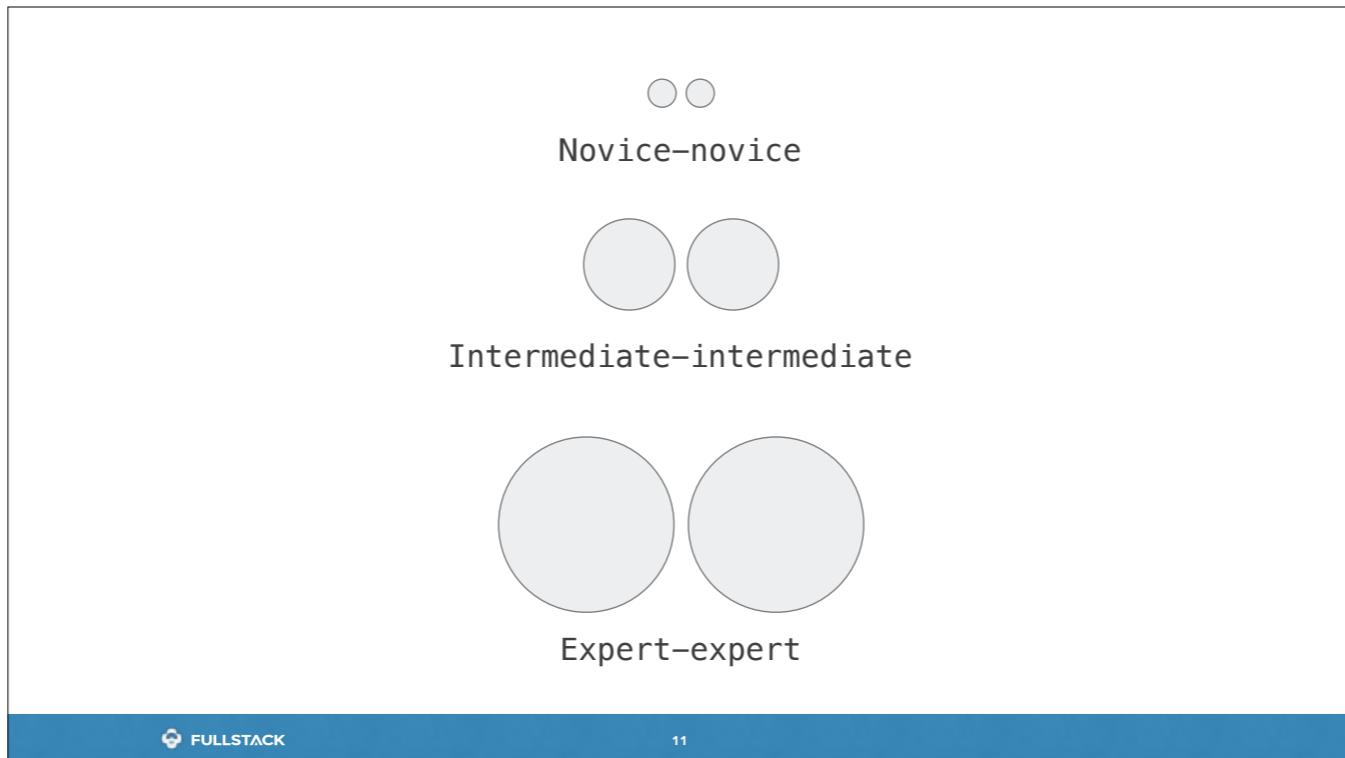
Pair Programming

Pair programming maximizes outcomes for teams where size == 2



Let's bucket developers into three simplistic groups:

Novice, intermediate, and expert developers



And let's consider three pairing scenarios.

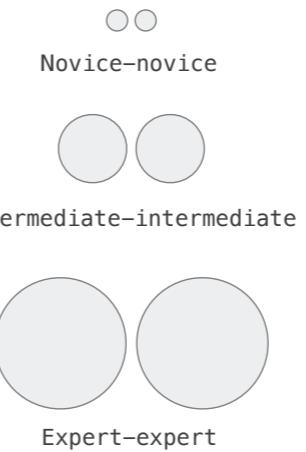
In these scenarios, we're pairing developers of roughly equivalent experience.

Let's Pair Up Right Now

Discuss the following:

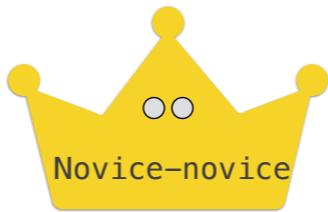
Which pair will see the least/most improvement
vs a solo developer with the same experience?

Why do you think that?



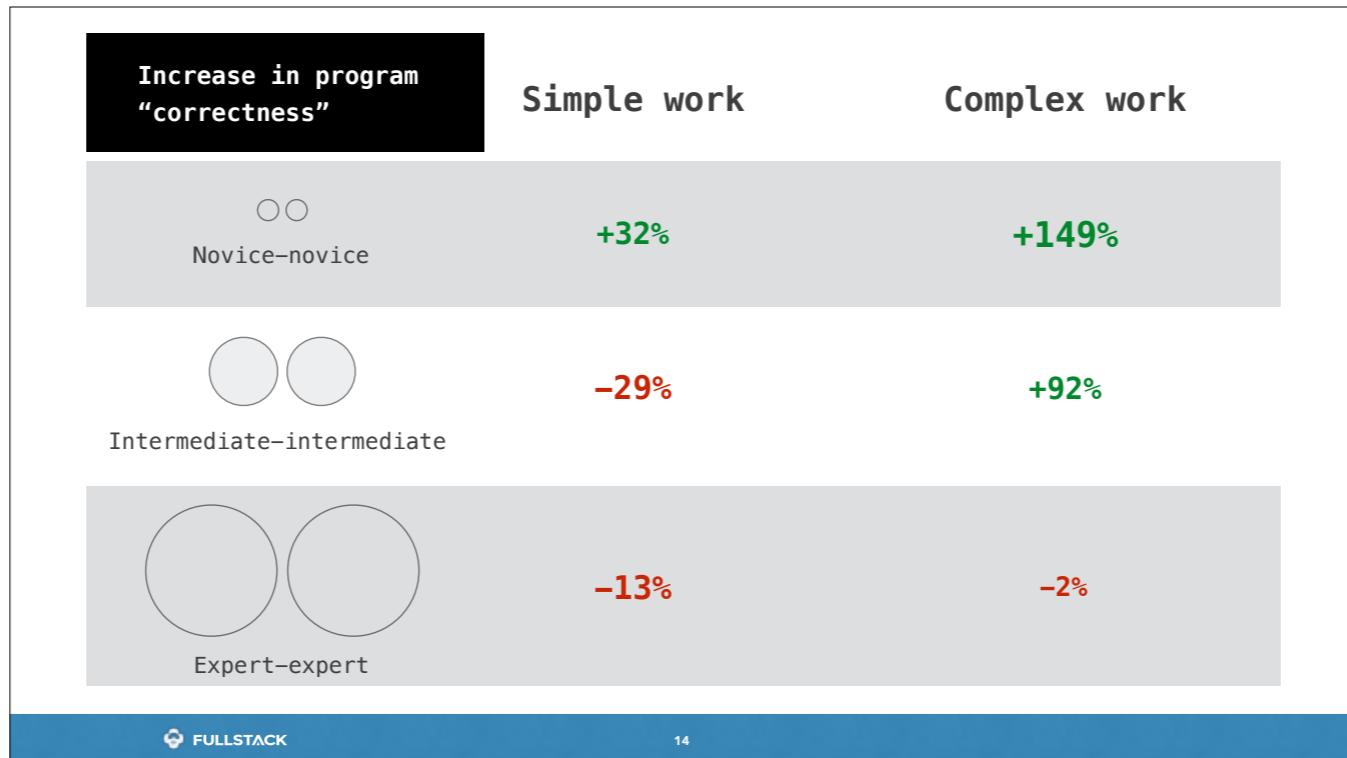
Have students group off with their neighbors (groups of two or three depending on the headcount)

Set a timer ~2 minutes for them to consider this question.



The effectiveness of pair programming: A meta-analysis (2009)
<http://www.ic.unicamp.br/~wainer/outros/systrev/30.pdf>

Novice-novice pairs see the most benefit from pairing.



By a wide margin.

Caveat: “correctness” sounds difficult to pin down for quantitative analysis :shrug:

Increase in
"correctness"

Novice-j

Intermediate-j

Expert-j

UberFacts @UberFacts

A study found that, when sober, rats prefer silence, but on cocaine, they prefer jazz.

ex work

149%

92%

Alex @_vivid_dreams

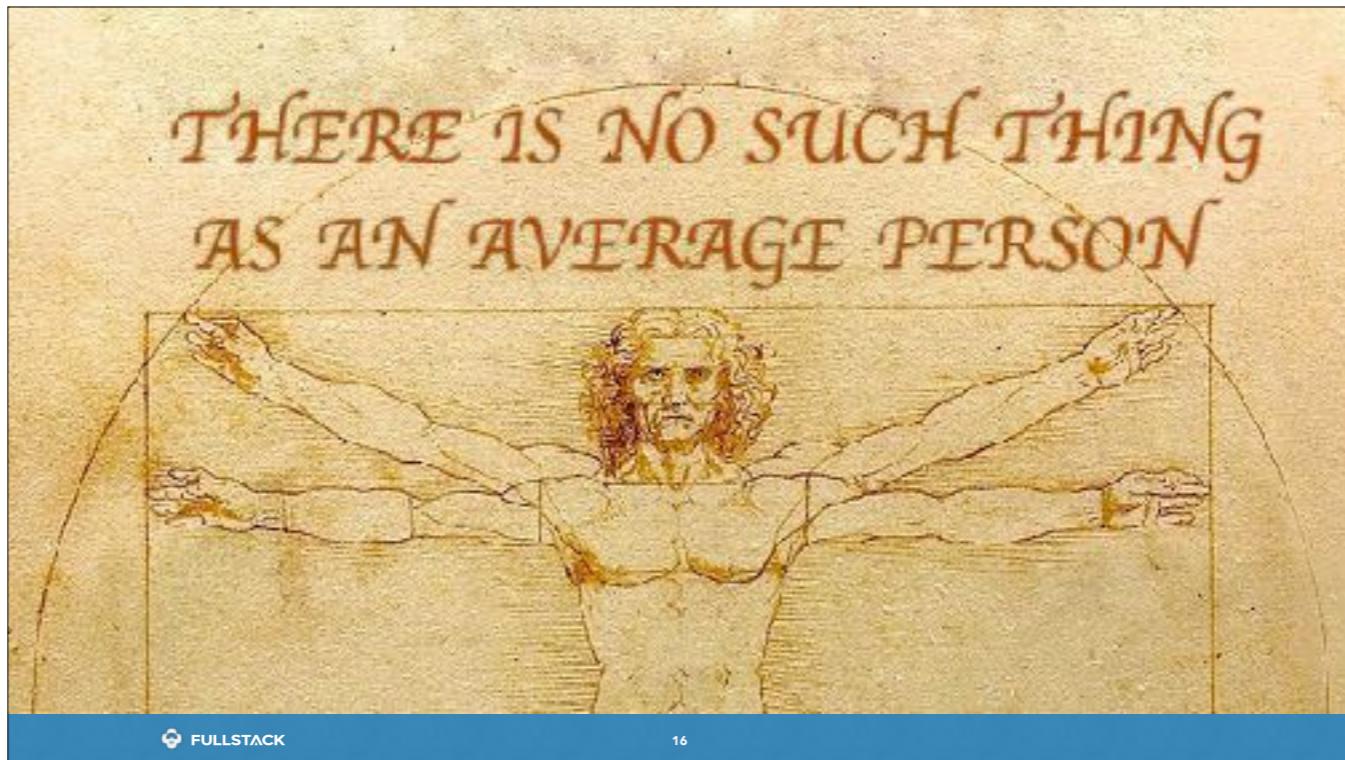
What do we do with this information

-2%

FULLSTACK

15

What do we do with this information?



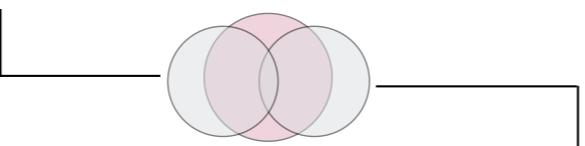
First thing is to remember, there is no such thing as an average person.

Air Force story: In the 1926, the US Air Force took the measurements of hundreds of pilots. The average of those measurements was used to design the cockpit of the aircraft.

You are not broken if you don't get a **%149** increase in correctness when you pair.

This analysis might be exposing the complexity of the problems under examination...

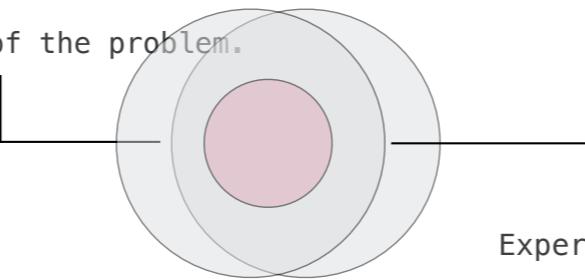
The difficulty of the problem.



Novice–novice pair

Novice–novice pairs might have a smaller overlap in know-how.
Working together they have more opportunities to lift each other up.

The difficulty of the problem.



Expert-expert pair

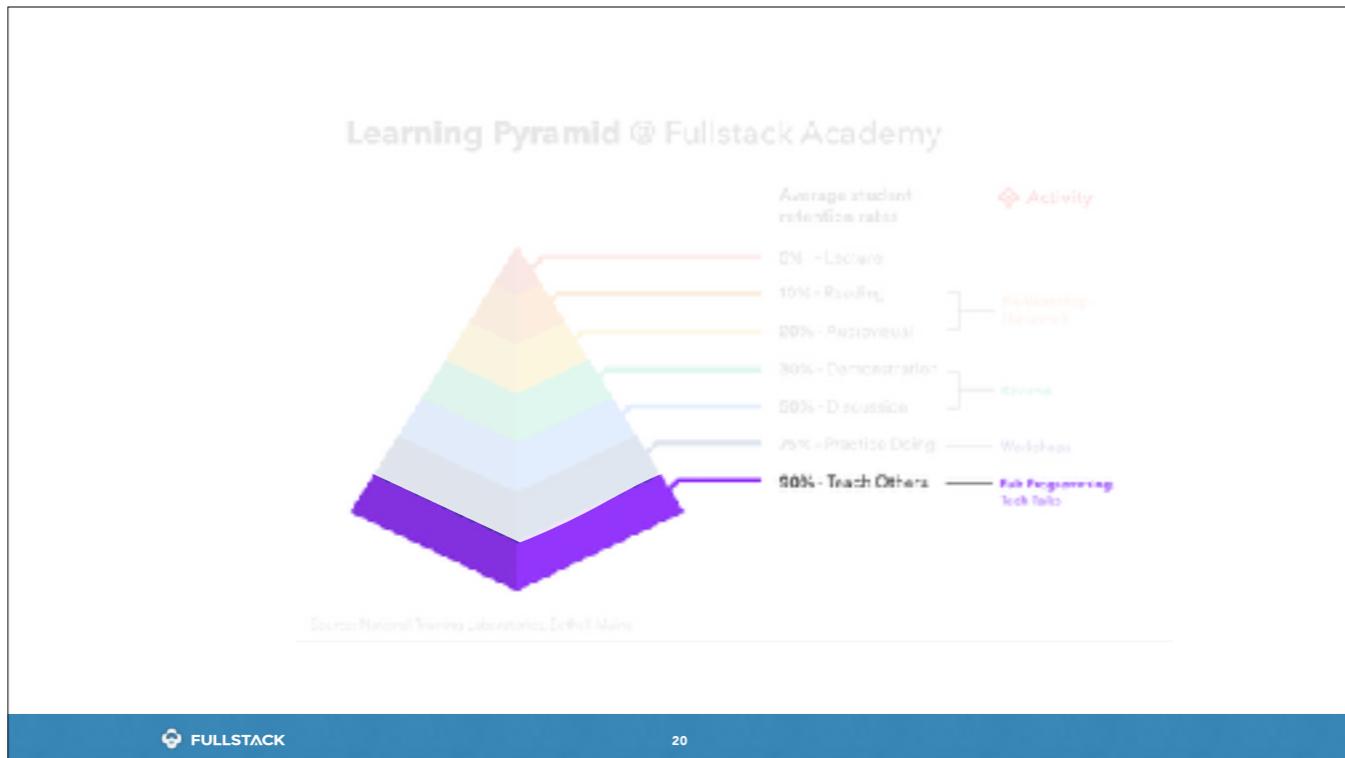
Expert-expert pairs might be individually capable of correctly solving the problem.
They might not have much of an opportunity to help each other out.

“If your brain is involved more than your hands
having another brain is helpful.”

— Tess M. 1707

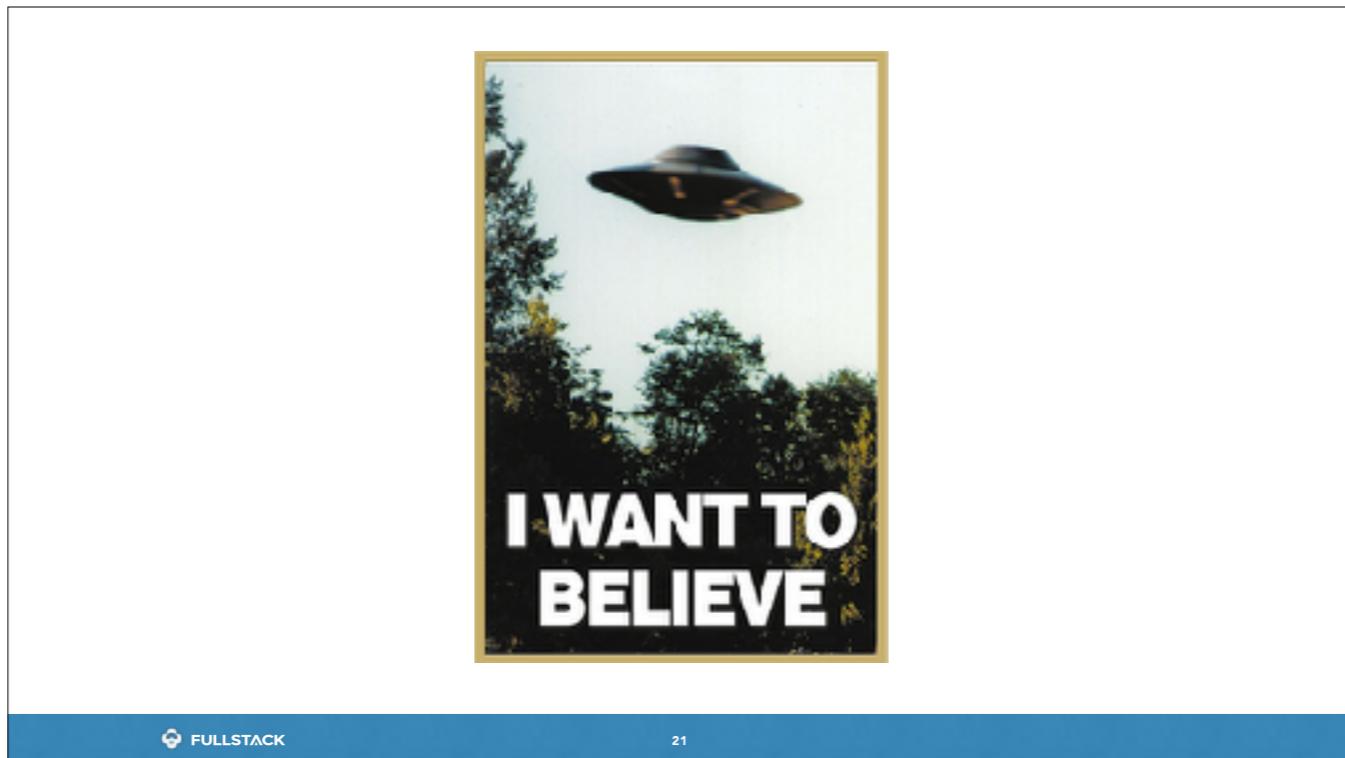


This sentiment was echoed in interviews with staff members.



This is part of why we value pair programming so highly at Fullstack Academy.

Remember this image from day-1 orientation?



So, now that you're a 100% true believer in pair programming, you may have a few questions, chief among them: **How do you actually pair program?**



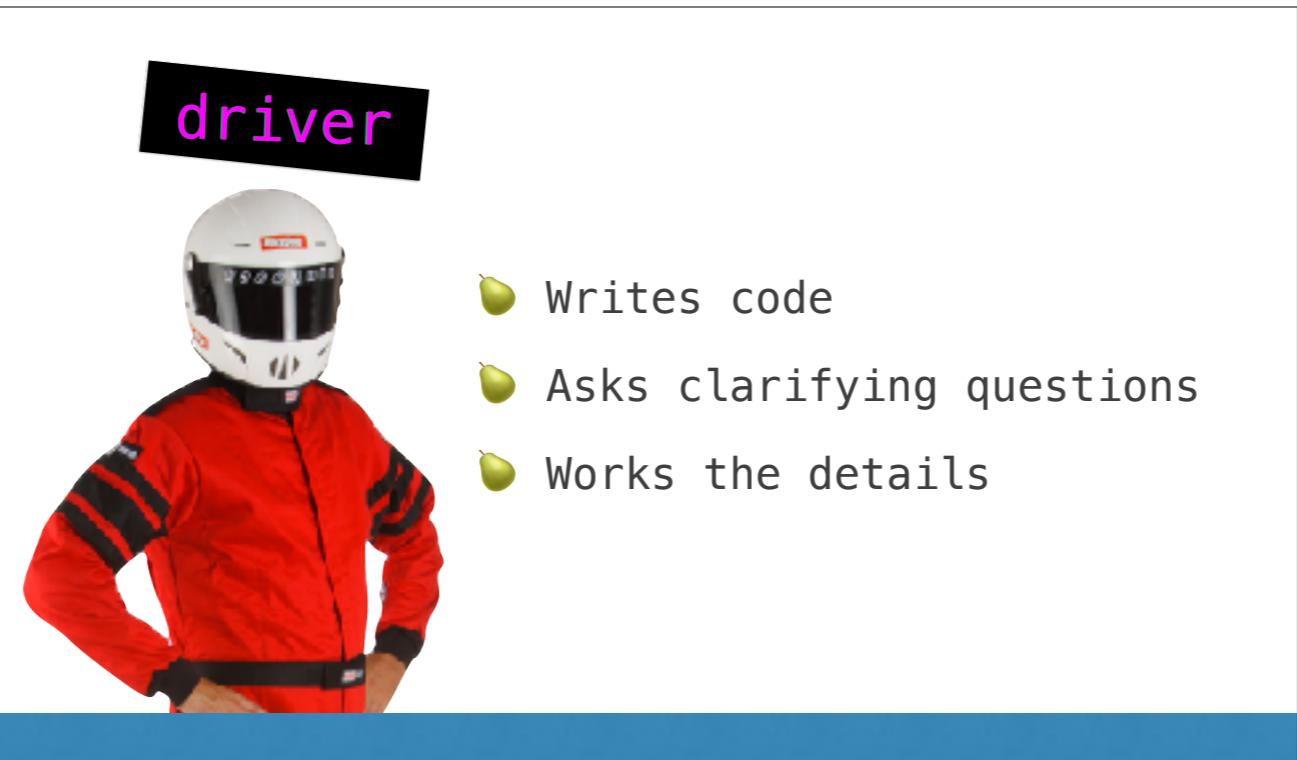
One way to think about pair programming is that one person is the driver, and the other is the navigator.

navigator

- 🍐 Spots syntax issues
- 🍐 Reads the docs
- 🍐 Sees the big picture
- 🍐 Speaks at a high level



Roles



- pear Writes code
- pear Asks clarifying questions
- pear Works the details

Roles



unbalanced
pairs

<https://www.flickr.com/photos/s2ublack>

Look at these kids on the see saw
Nobody is having any fun

Dance of the Little Swans



Consider the Dance of the Little Swans from Swan Lake

Not only do they have to dance
They have to dance together

During rehearsal, it could be easy to become frustrated with a dancer who isn't picking
up the choreography as fast.

The duty is assigned to the group, hand-in-hand. The audience will consider their success collectively.

Tie your perception of success to your pair, rather than your own success as an individual. You may be able to get through a Fullstack workshop on your own, but you'd be throwing away an opportunity to practice
programming as a team sport dance.

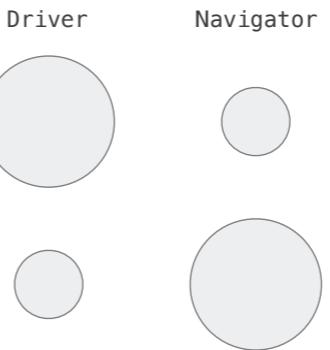
Let's Pair Up Right Now

Discuss the following:

How might it effect the pairing session if the participants have different experience levels?

How is it different if the Driver or the Navigator is the stronger participant?

What can be done to mitigate these issues?



Have students pair up with their neighbors
(groups of two or three depending on the headcount)

**Set a ~5 minute timer for them to consider these questions.
Then discuss.**

 DETRIMENTAL
NAVIGATOR 

-  Grabs the keyboard
-  Dictates code
-  Checks out





DETRIMENTAL
DRIVER



- 💀 Speeds ahead
- 💀 Ignores navigator
- 💀 Won't revisit code



**EXCELLENT
NAVIGATOR**



- 🍐 Diagrams the approach
- 🍐 Confirms understanding
- 🍐 Stays engaged





GREAT
DRIVER



- 🍐 Narrates their typing
- 🍐 Confirms understanding
- 🍐 Keeps steady pace



Don't forget, you're in the same car, barreling down a hillside at 80mph.

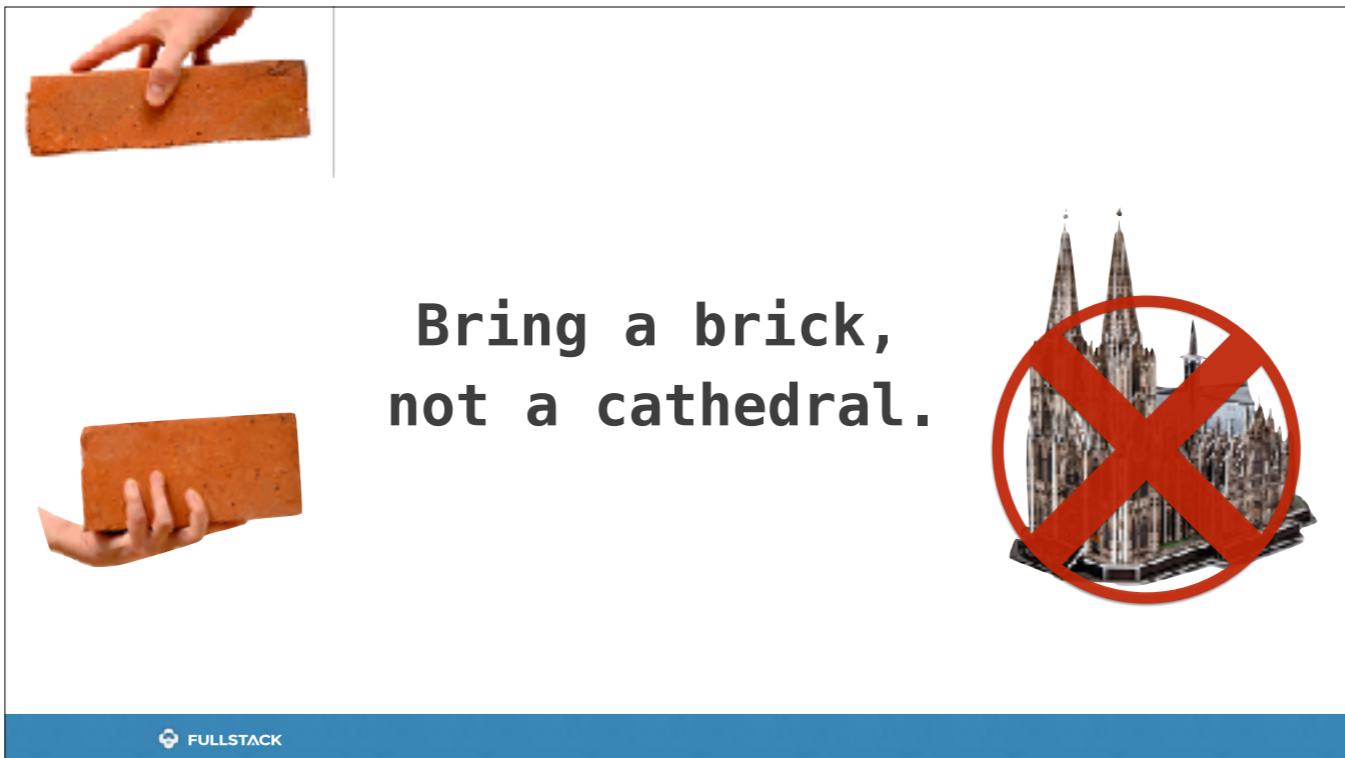
Decisions and approach are ultimately decided upon as a team.

One set of hands on the keyboard



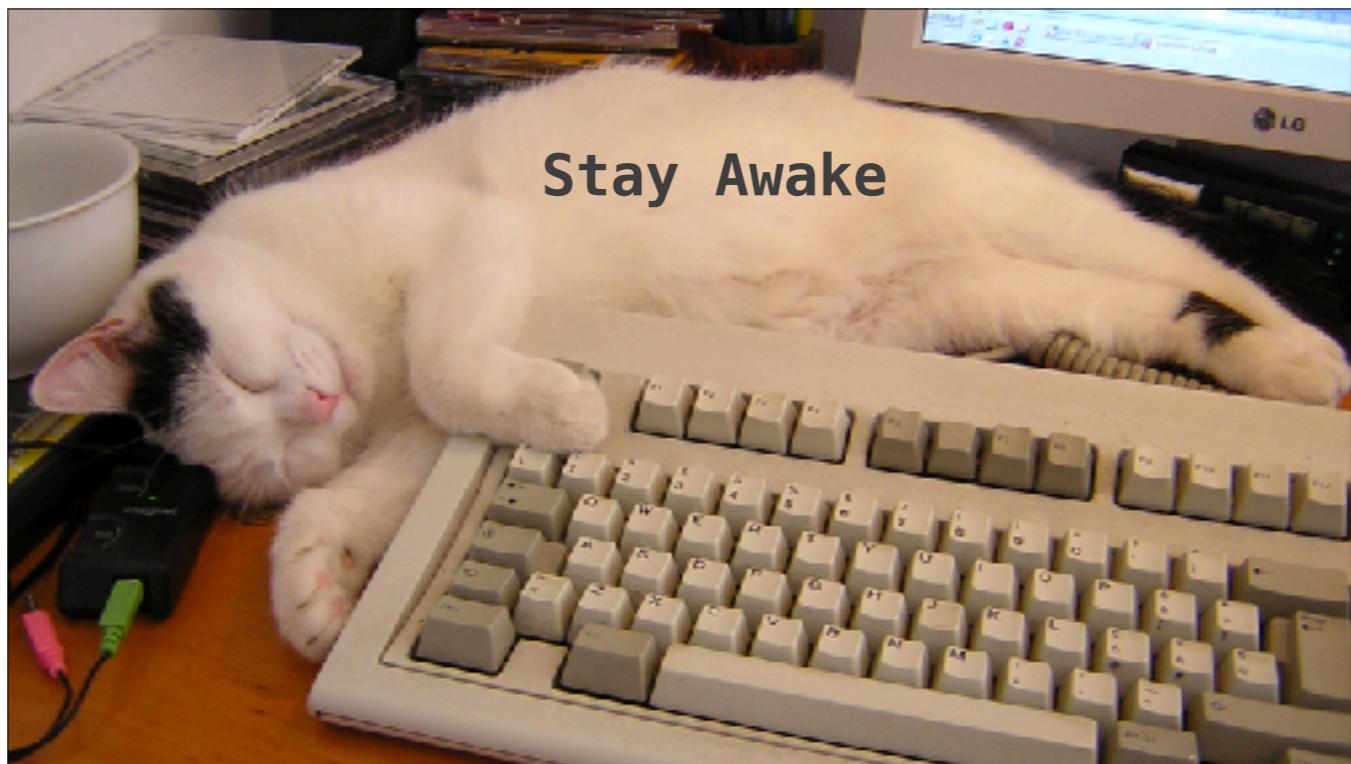
FULLSTACK

Driver has hands on keyboard, does the literal programming.
Navigator is keeping track of the bigger picture. Seeing the curves in the road ahead.



Both parties are making decisions about higher level approach.

Being the Driver is not an excuse to check-out on the approach.
Being the Navigator is not an opportunity to steamroll your partner.



Stay Awake

Things You Learned



Pairing combines your minds into one super-mind.



How to navigate

How to drive



Specific instructions on how to work together effectively when things aren't going well.

