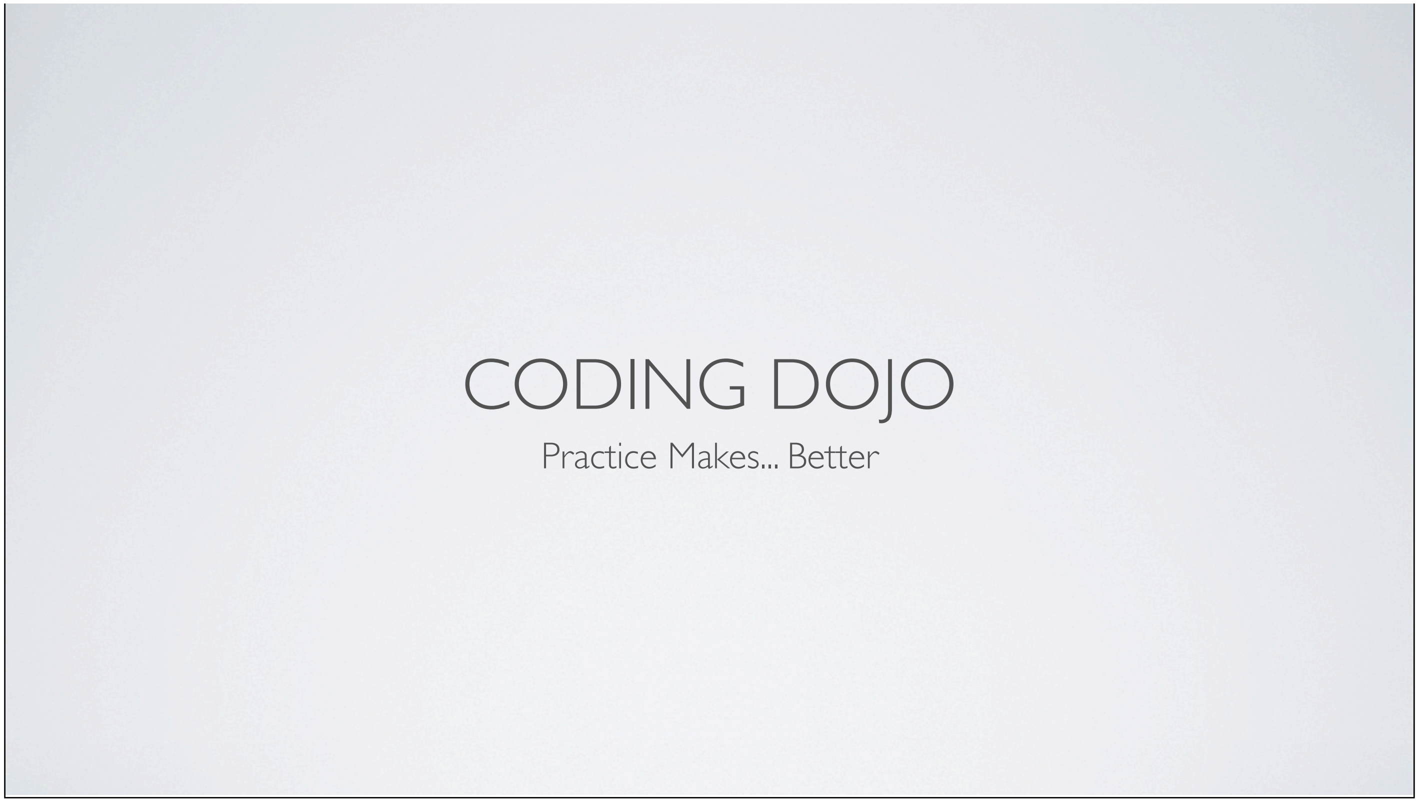
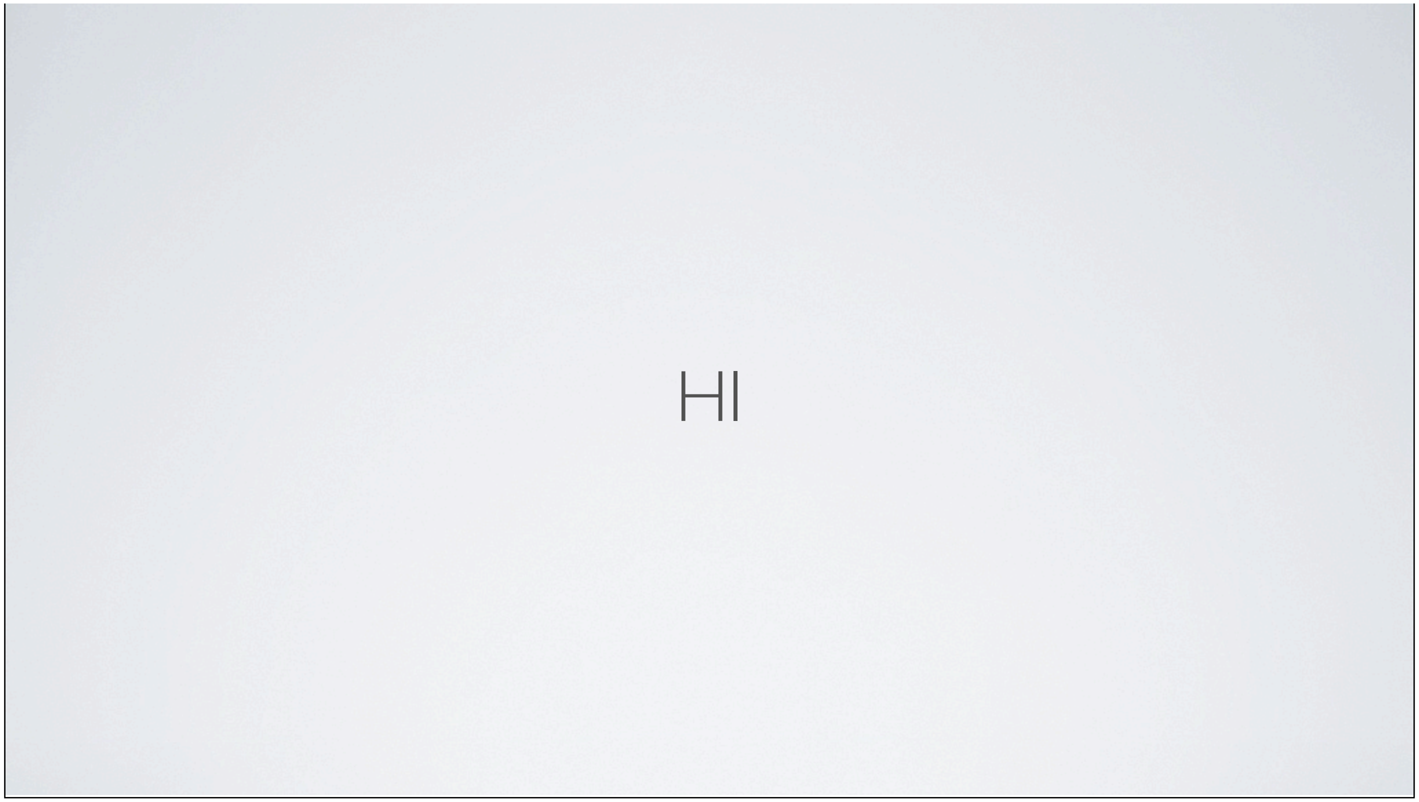


CODING DOJO

Practice Makes... Better

Welcome to the Coding Dojo.

I just wanted to go over a couple of things before we get to the dojo.

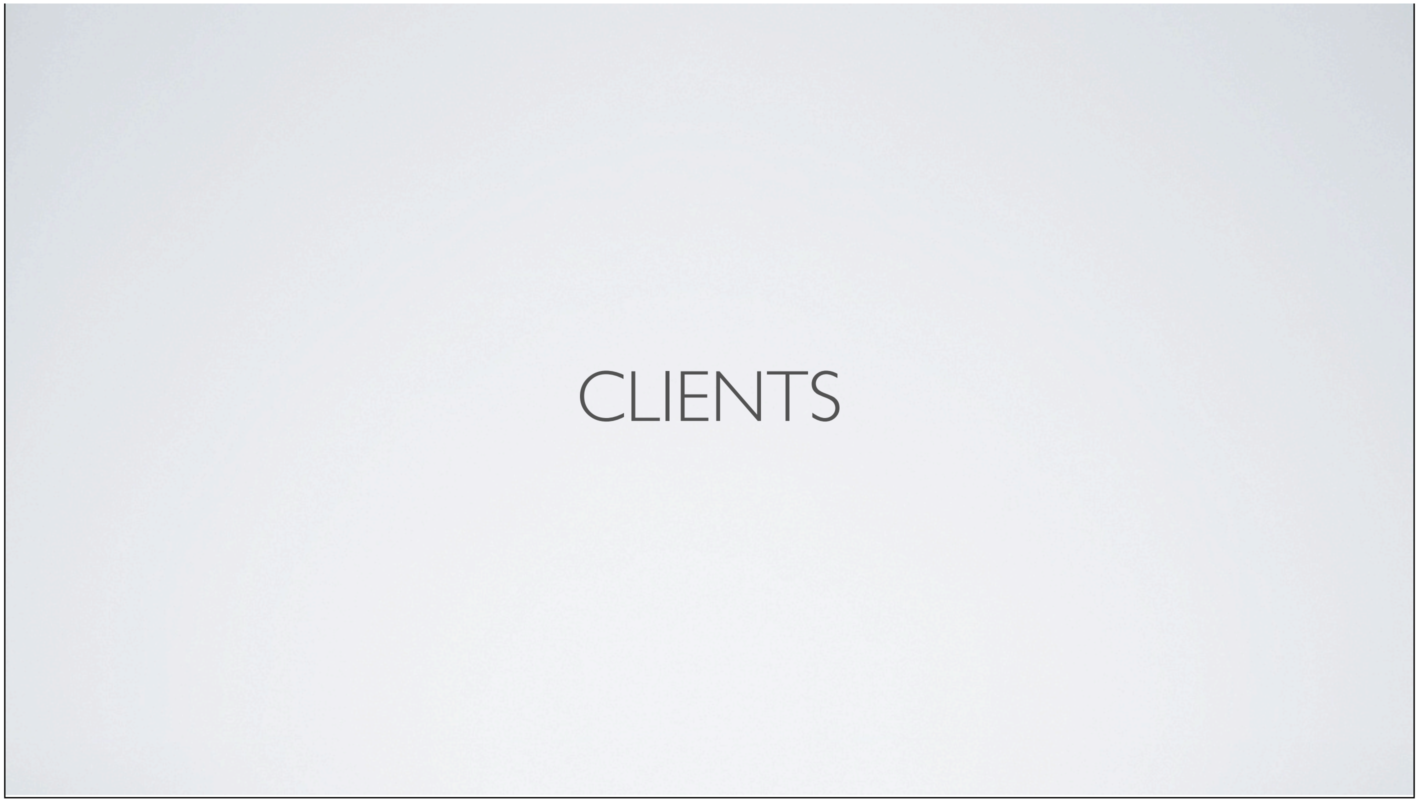


HI

- Name
- Graphic Designer turned Web Designer turned Freelance Ruby on Rails Developer
- In my career I’ve seen all levels of the dev process

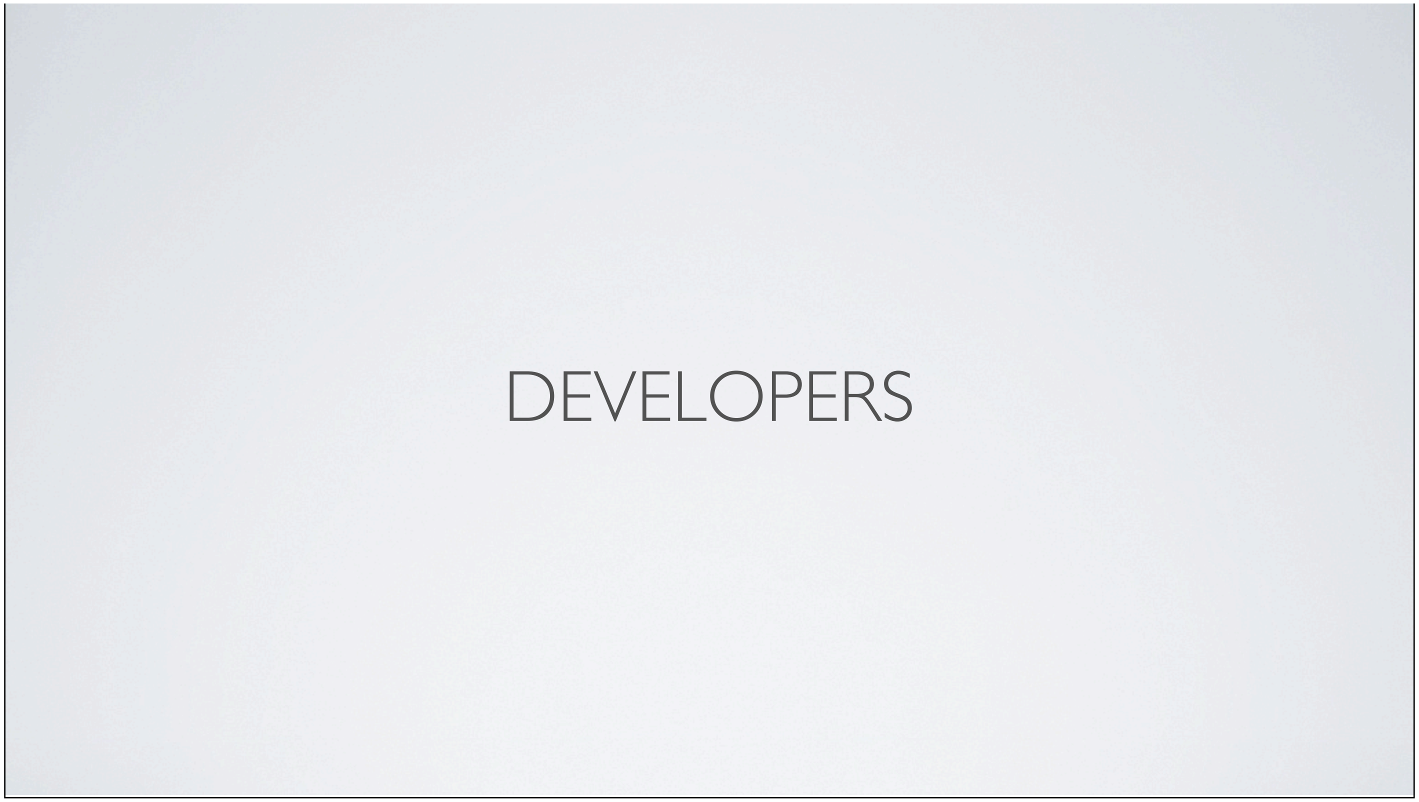
    initial client meetings
    design phases
    developer management
    development

- Became interested in testing in because I’ve written faulty code.
- But there is are more important benefits to testing than just making sure your code doesn’t break.



CLIENTS

- Market-speak and buzzwords
- Don’t know what they want
- Change their mind all the time
- Features are bullet points in 30 page RFPs
- Seeking fads instead of business value



DEVELOPERS

- Speak a different language
- Terrible estimators—everything is possible and easy
- Loners and Heroes
- My codez are perfect

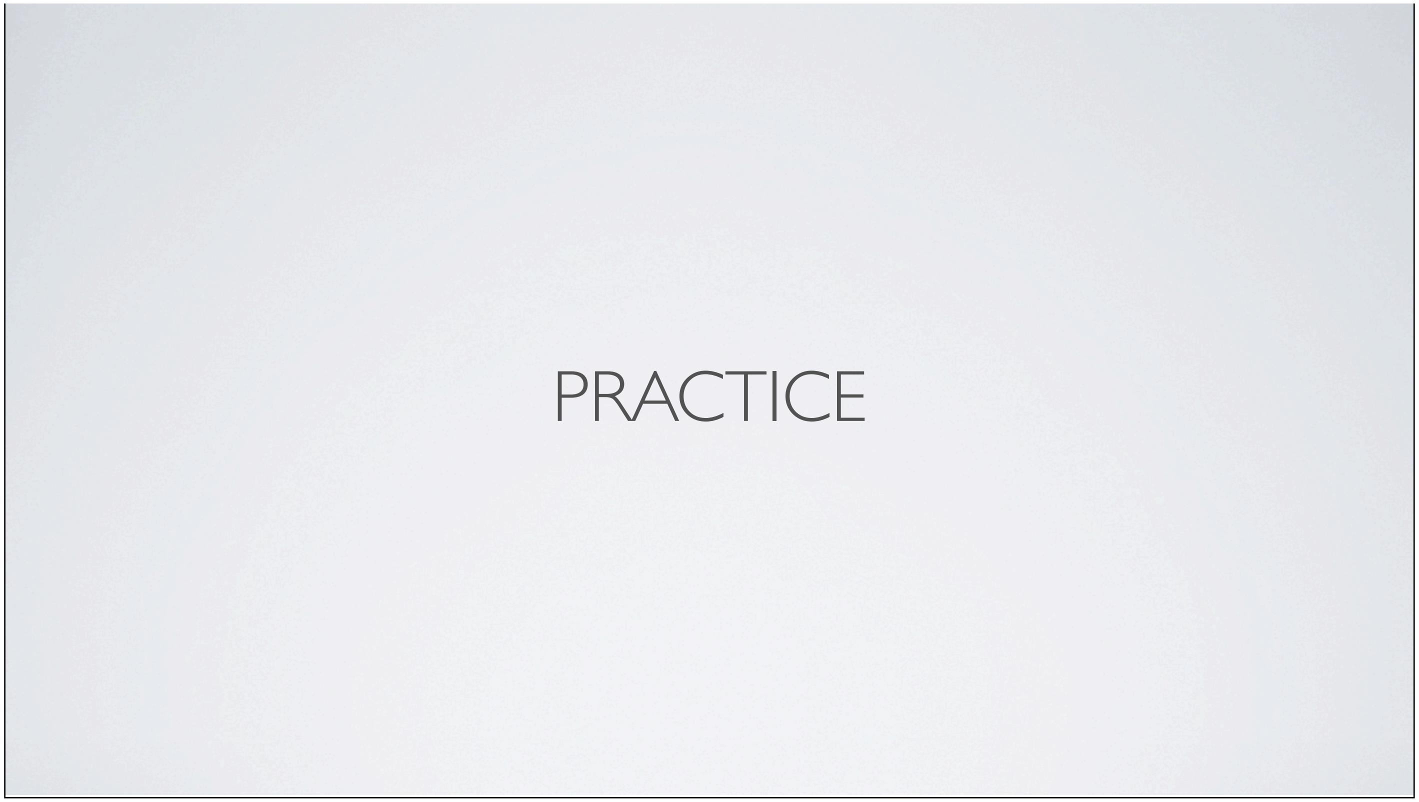
- more than one way to do something
- ruby devs are largely newbs

COMMUNICATION

wouldn’t it be great if we could all speak the same language? English!
wouldn’t it be great if that language was executable! Yes!

Cucumber:
focus on business value
align client’s wishes / expectations with developer goals and outcome
test the outside of your app to the inside

RSpec:
focus on inter-team communication
test the inside of your app’s objects



PRACTICE

Testing is hard.
Repetition of the process over outcome.
Start small, and build to larger, harder problems.

CODING DOJO

- A pair at the keys at a time. One person writes specs, the other writes implementation
- Both the tester and the implementor should describe what they're doing / thinking aloud.
- Baby steps! The smallest amount of code to make the test pass as possible
- The group can offer suggestions for refactorings on a green bar.
  The pair can request silence on a red bar.
- Every 5 minutes, the tester should swap out with a new person,
  and the implementor should become the tester

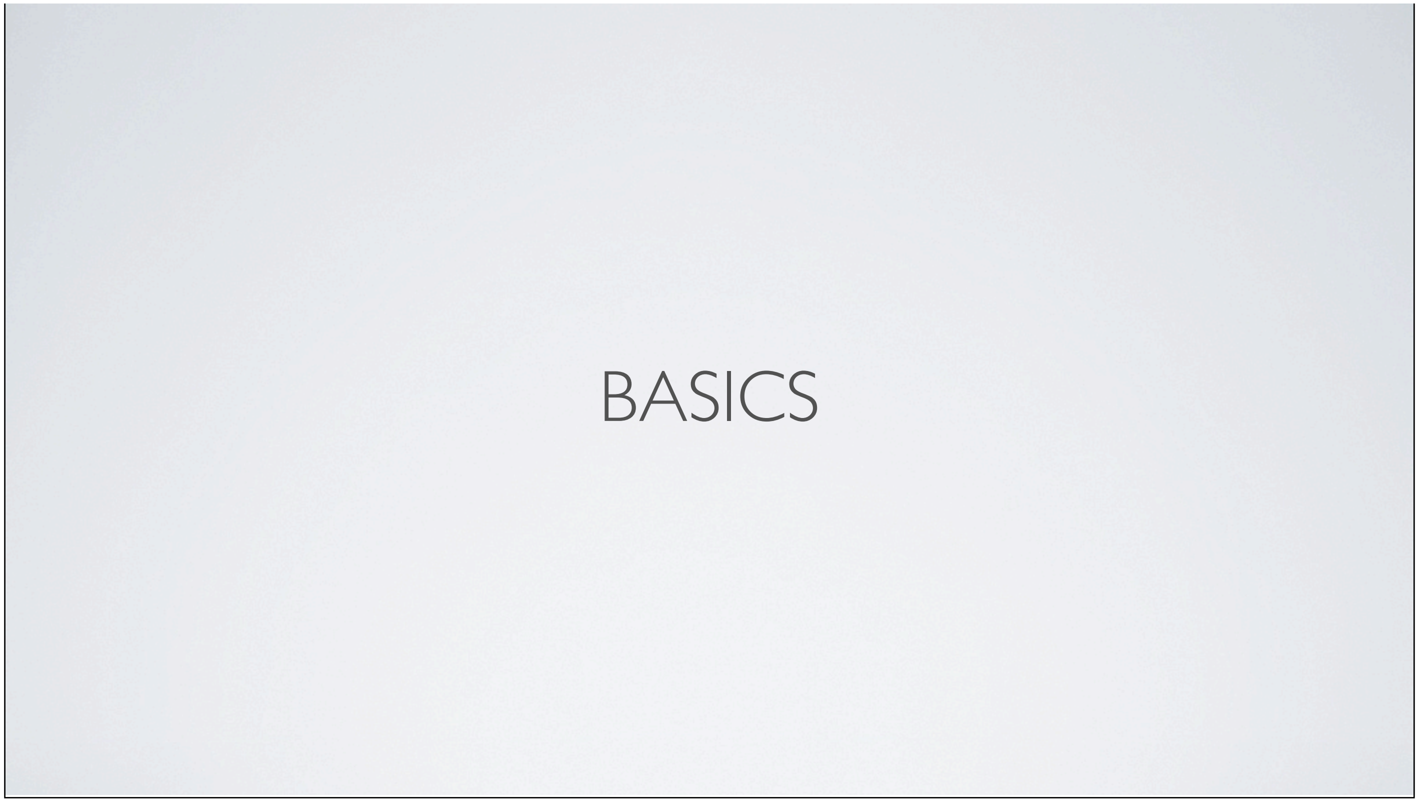
RULES OF BOWLING

- There are 10 frames in a game. Each one provides two opportunities for a player to knock down 10 pins.

- Frame is marked “strike” if the player hits 10 pins on the first roll. To score a strike, you add the next two rolls to the current frame’s pins (10). Example: Rolls = 10, 1, 1, Frame 1 score is 12, Frame 2 score is 2.

- Frame is marked “spare” if the sum of the player’s 2 rolls is 10. To score a spare, you add the next 1 roll to the current frame’s pins (10). Example: Rolls = 5, 5, 1, 0, Frame 1 score is 11, Frame 2 score is 1.

- A spare on the tenth frame allows 1 bonus roll. A strike on the tenth frame allows 2 bonus rolls.



BASICS

Open up Code window to show basics

project setup and layout

cucumber

- Plain text input through Ruby bridge
- write the code you wished you had

rspec

- relationships (person.should have(2).parents)
- predicates (person.should be_awesome => person.awesome?)
- properties (person.full_name.should == 'Dave Giunta')

red green refactor

THANKS!

Dave Giunta
dgiunta@mac.com
dgiunta on twitter and github

Coding Dojo Files:
http://github.com/dgiunta/chicagoruby.org-4-18-coding-dojo/