What is good code?
Evaluating code quality

Daniel Carral
Barcelona on Rails (@ XING), 02/15/2018

github.com/dcarral/good-code

Agenda

➔ **Intro**

➔ **Craft**
Let's solve the problem!

➔ **Show & discuss**
Which approach did you use? Why?

➔ **Evaluate**
Opinion vs facts
What do the numbers say?
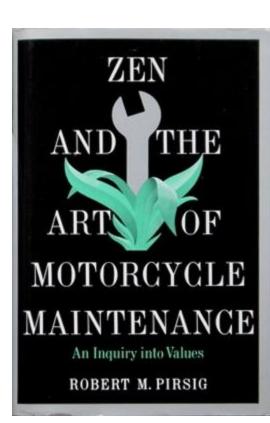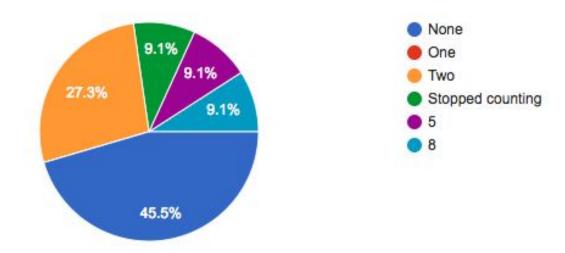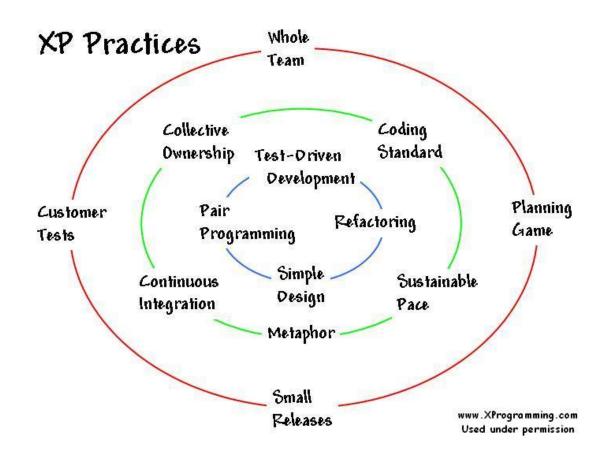
➔ **Retrospective**

What is good?

Good?

Good?

Good?
Better?

ZEN AND THE ART OF MOTORCYCLE MAINTENANCE

An Inquiry into Values

ROBERT M. PIRSIG

Craft
Coding Dojo

How many dojos?

- None
- One
- Two
- Stopped counting
- 5
- 8

45.5% None
27.3% Two
9.1% Stopped counting
9.1% 5
9.1% 8

extreme Programming explained

EMBRACE CHANGE

Kent Beck

XP Practices

Whole Team · Coding Standard · Planning Game · Small Releases · Customer Tests · Collective Ownership · Continuous Integration · Test-Driven Development · Metaphor · Sustainable Pace · Pair Programming · Simple Design · Refactoring

www.XProgramming.com
Used under permission

TDD & refactoring

Pair programming

Simple design

Priority ↑

Passes the Tests

Reveals Intention

No Duplication

Fewest Elements

Kata
99 bottles of beer

Clone & code :)
github.com/dcarral/99bottles-polyglot

35 mins to go

Clone & code :)
github.com/dcarral/99bottles-polyglot

Time over :)

Show & discuss

(or not ;)

How did you solve it? Why?

Reminder

The goal is not to judge our coding skills.
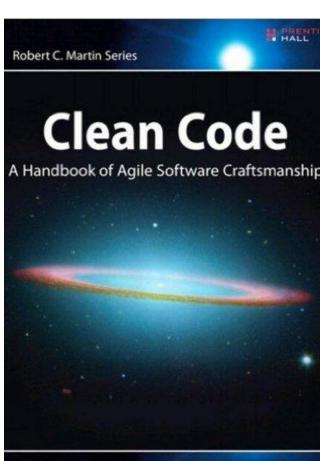
We're discussing how to **evaluate code quality.**

github.com/dcarral/99bottles-solutions

Agenda

➜ **Intro**

➜ **Craft**
Let's solve the problem!

➜ **Show & discuss**
Which approach did you use? Why?

➜ **Evaluate**
Opinion vs facts
What do the numbers say?

➜ **Retrospective**

Evaluating
code quality
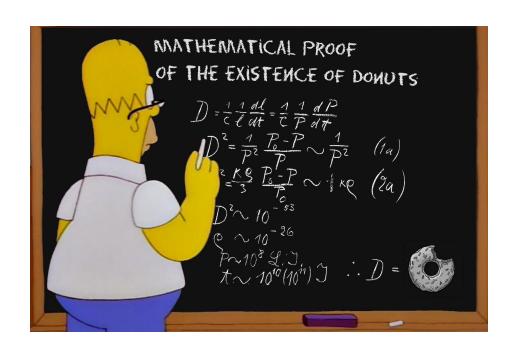
Based on opinion

Clean Code

A Handbook of Agile Software Craftsmanship

Robert C. Martin

In your opinion, what is "good code"?

- "Easy to understand and read. Easy to test."

- "SOLID"

- "Passes the squint test :)"

- "Quality transcends description"

Based on facts

Software metrics

"A software metric is a standard measure of a degree to which a software system possesses some property."

Why measure?

"You cannot control what you cannot measure."

— Tom DeMarco

Some metrics

- Ca 1960: LOC

- 1976: Cyclomatic complexity

- 1997: ABC size

ABC

- Assignments

- Branches (of control)

- Conditions

```
class Test
  def blah
    a = eval "1+1"
    if a == 2 then
      puts "yay"
    end
  end
end
```

Is seen by flog as:

```
class Test
  def blah           # 11.2 =
    a = eval "1+1" # 1.2 + 6.0 +
    if a == 2 then # 1.2 + 1.2 + 0.4 +
      puts "yay"   # 1.2
    end
  end
end
```

and reported as:

```
Test#blah: (11.2)
    6.0: eval
    1.2: branch
    1.2: ==
    1.2: puts
    1.2: assignment
    0.4: lit_fixnum
```

"Confessions of a Ruby Sadist"

4 more solutions

What do the numbers say?

Solution	LOC	Flog total	Flog worst bit
#1 Incomprehensibly concise	19	42.5	36.2 (#verse)
#2 Speculatively general	63	54.8	28.5 (lambdas)
#3 Concretely abstract	92	67	15.3 (#challenge)
#4 Shameless green	34	25.6	19.3 (#verse)

Why metrics?

*"Metrics are fallible but human opinion is no more precise. Checking metrics regularly will **keep you humble** and __**improve your code**__."*

— Sandi Metz, Katrina Owen

Ruby Edition

99 Bottles of OOP

A Practical Guide to Object-Oriented Design

Sandi Metz & Katrina Owen

sandimetz.com/99bottles/postcard

Beautiful Code

Leading Programmers Explain How They Think

O'REILLY®

Edited by Andy Oram & Greg Wilson

Retro time

- Coding dojos
- TDD
- Refactoring
- Pair programming
- Simple design

- *Zen & the art of motorcycle maintenance*
- *Clean code*
- *XP explained*
- *99 bottles of OOP*
- *Beautiful code*

- Software metrics
- LOC
- Cyclomatic complexity
- ABC size

@dcarral

dcarral.org