What is good code? Evaluating code quality

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







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

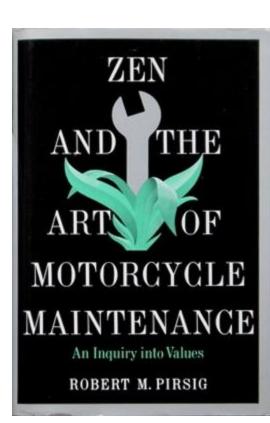
What is good?





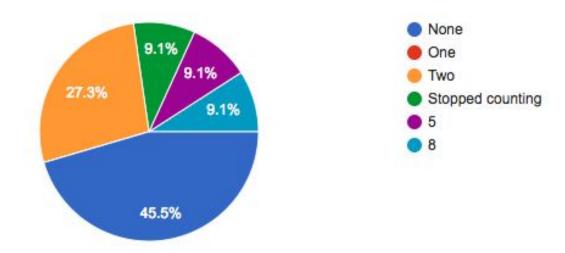






Craft Coding Dojo

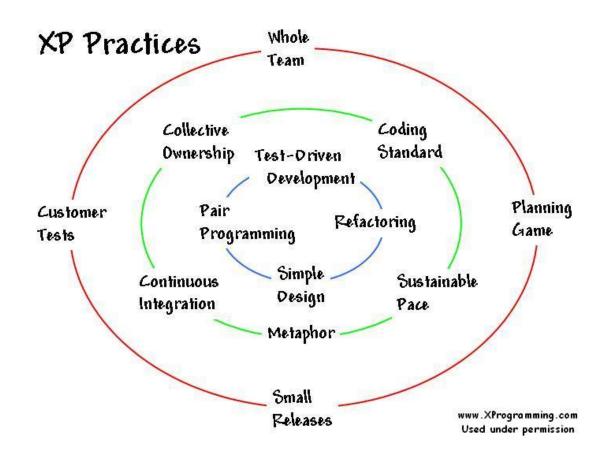
How many dojos?





EMBRACE CHANGE

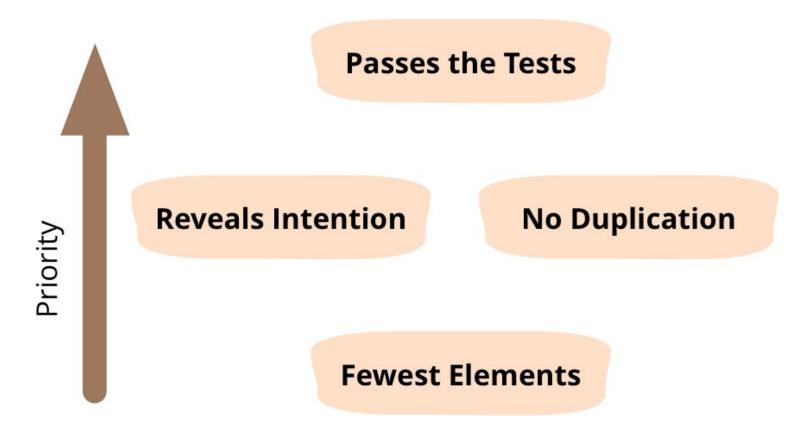
Kent Beck



TDD & refactoring

Pair programming

Simple design



Kata 99 bottles of beer



(until 20:30h)

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



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.





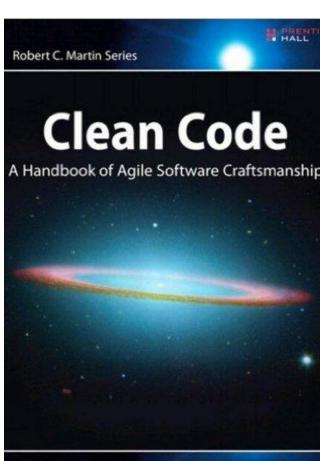
Agenda

- → 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





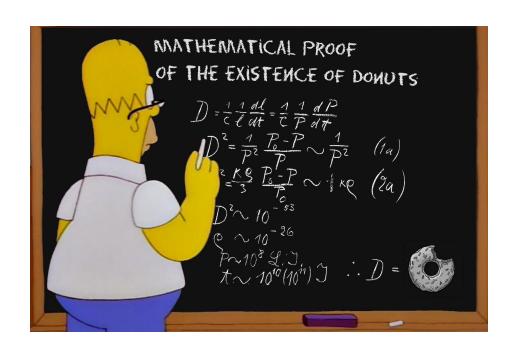
Foreword by James O. Coplien

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

* Apparently, he regrets having said this (thanks David!)

Some metrics

• Ca 1960: LOC

1976: Cyclomatic complexity

• 1997: ABC size

ABC

Assignments

Branches (of control)

Conditions

```
class Test
                                        class Test
 def blah
                                          def blah
                                                           # 11.2 =
   a = eval "1+1"
                                            a = eval "1+1" # 1.2 + 6.0 +
                                            if a == 2 then # 1.2 + 1.2 + 0.4 +
   if a == 2 then
                          Is seen by
                                              puts "yay" # 1.2
      puts "yay"
                           flog as:
   end
                                            end
 end
                                          end
end
                                        end
                                        Test#blah: (11.2)
                                             6.0: eval
                                             1.2: branch
                                             1.2: ==
                         and reported
                                             1.2: puts
                             as:
                                             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

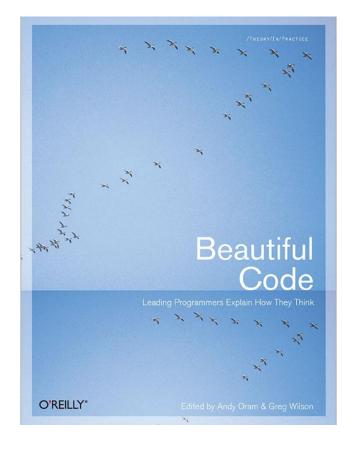
99 Bottles of OOP

A Practical Guide to Object-Oriented Design



Sandi Metz & Katrina Owen

sandimetz.com/99bottles/postcard



* Disclaimer: Some people don't like it (at all). Think twice before purchasing it;)

Retro time

... surprised me. I've learnt... I'd like to start Feedback:) doing...



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

- Software metrics
- LOC
- Cyclomatic complexity
- ABC size

- Zen & the art of motorcycle maintenance
- Clean Code
- eXtreme Programming explained
- 99 bottles of OOP
- Beautiful code

@dcarral dcarral.org