CLEAN CODE

PRINCIPLES, PATTERNS, AND PRACTICES FOR BETTER CODE

Jeremy Clark & Theo Jungeblut

WHY WE'RE HERE

Separation of Concerns

SRP

KISS

Constructor Injection

Readability

YAGNI

OCP

Refactoring

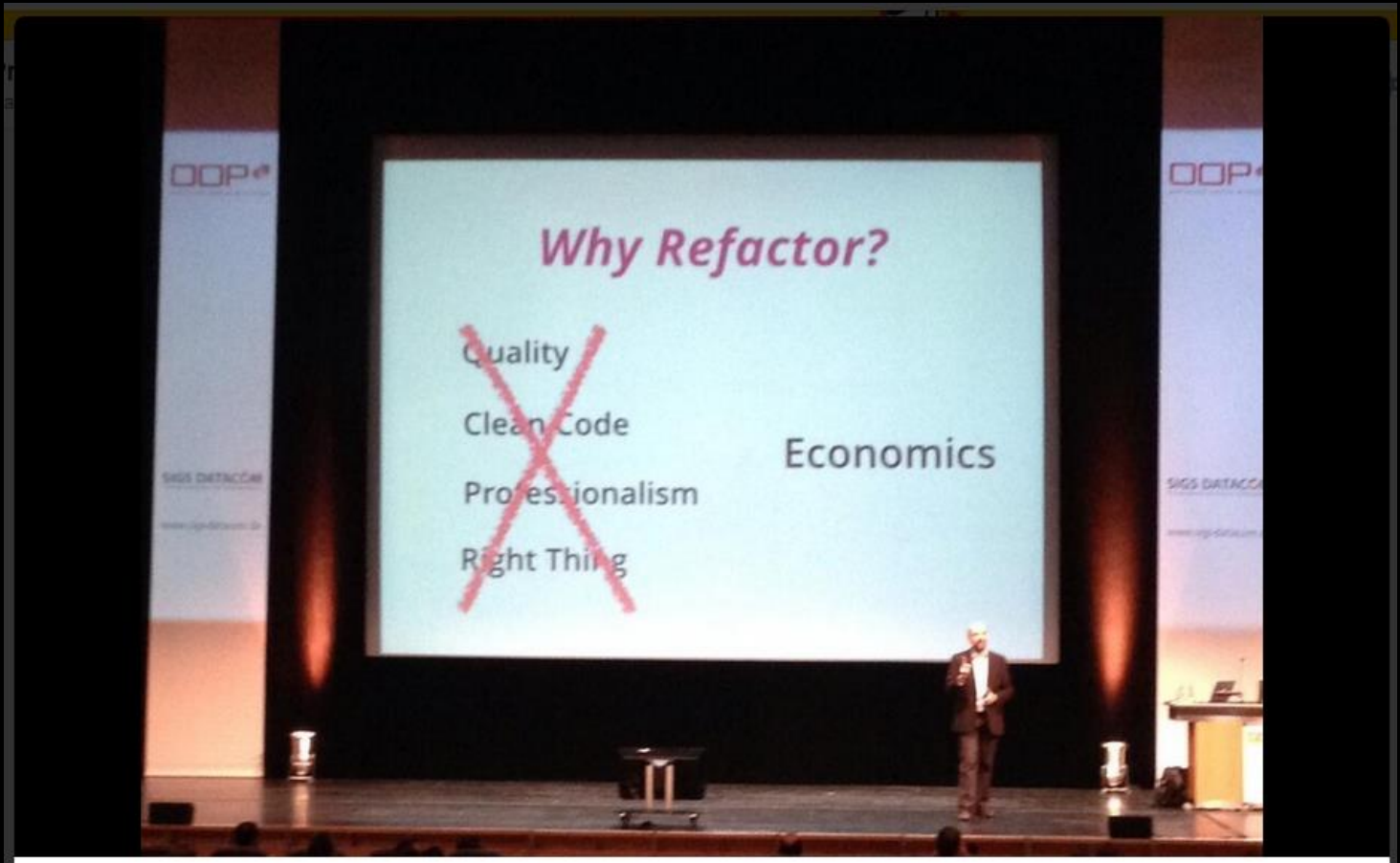
Testing

DRY

Naming

Design Patterns

THE REAL REASON



About Programming @abt_programming · Apr 15

"If you use the arguments on the left to justify refactoring, you're screwed" - Martin Fowler via @mbohlende pic.twitter.com/np4uQk1b95

Reply Retweet Favorited Flag media

DOES WRITING

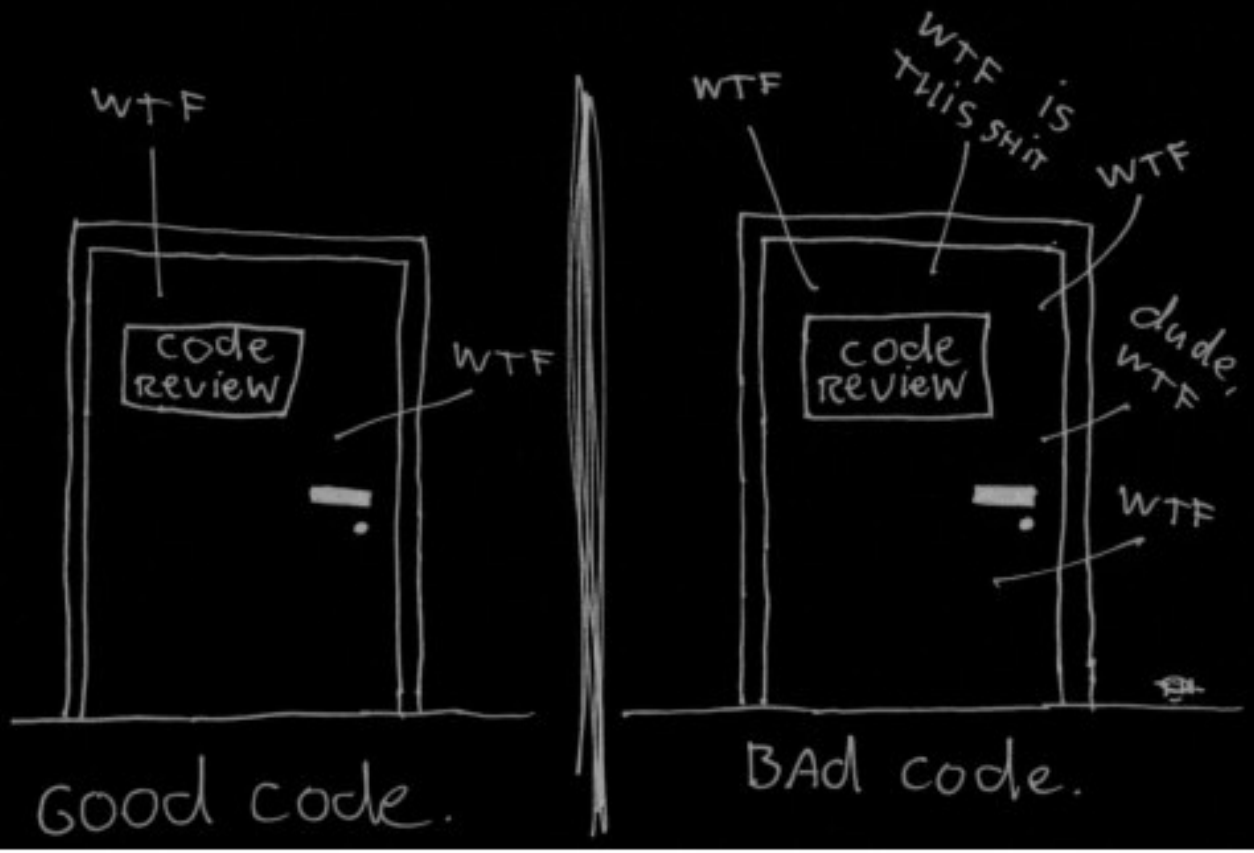
CLEAN CODE

MAKE US

MORE EFFICIENT?

WHAT
IS
CLEAN CODE?

The only valid measurement of code quality: WTFs/minute



(c) 2008 Focus Shift/OSNews/Thom Holwerda - http://www.osnews.com/comics

WHY WE'RE HERE

Readable, Maintainable, Testable Code

- How to Get There
  - Best Practices
  - Loose Coupling
  - Separation of Concerns
  - S.O.L.I.D.
  - Unit Testing
  - Naming

LOGISTICS

- Restrooms
- Food
- Cell Phones
- Network

AGENDA

- 8:30 – Welcome
- 9:00 – Design Patterns
- 9:30 – Design Principles
- 10:00 – Break
- 10:15 – Dependency Injection
- 11:15 – Lab
- 12:00-ish – Lunch & Networking

AGENDA

- 12:45 – Craftsmanship
- 13:45 – Lab
- 14:15 – Break
- 14:30 – Lab
- 15:00 – Lab Review
- 15:30 – Summary / Raffle / Wrap Up
- 16:00 – Start Being Awesome

CLEAN CODE

PRINCIPLES, PATTERNS, AND PRACTICES FOR BETTER CODE

Next Up: Design Patterns