



Software Craftsmanship

```
fn whoiam() {  
  
    let lastname = "Moreau";  
    let firstname = "David";  
    let company = "Thales Alenia Space";  
    let email = "david.moreau@thalesaleniaspace.com";  
    let job = "Developer/LeadTech/ProductOwner/Craftsman";  
  
}
```



Agile Manifesto (2001)

Individuals and Interactions over
processes and tools

Working Software over
comprehensive documentation

Customer Collaboration over
contract negotiation

Responding to Change over
following a plan



TL;DR;

- Accept change
- Learn to ride
- Feedback loop providing data



Agility Everywhere

The Hangover

Inefficient Develop/
Debug/Deploy cycles

Unreliable and
costly tests

Late discovery
of bugs

Unstable system

Requirements
not well
understood

Stagnant
Skillset

Low Moral
and
Motivation

Mountain
of
Technical
Debt

Lack of
technical
expertise

Long
running
builds

Unreliable
Release
Process



What's Wrong?



**In a software product, the
most important
deliverable is the code
itself**



**Why better code is so
important?**

TECHNICAL DEBT

I DON'T
UNDERSTAND
WHY IT TAKES
SO LONG TO
ADD A NEW
WINDOW.

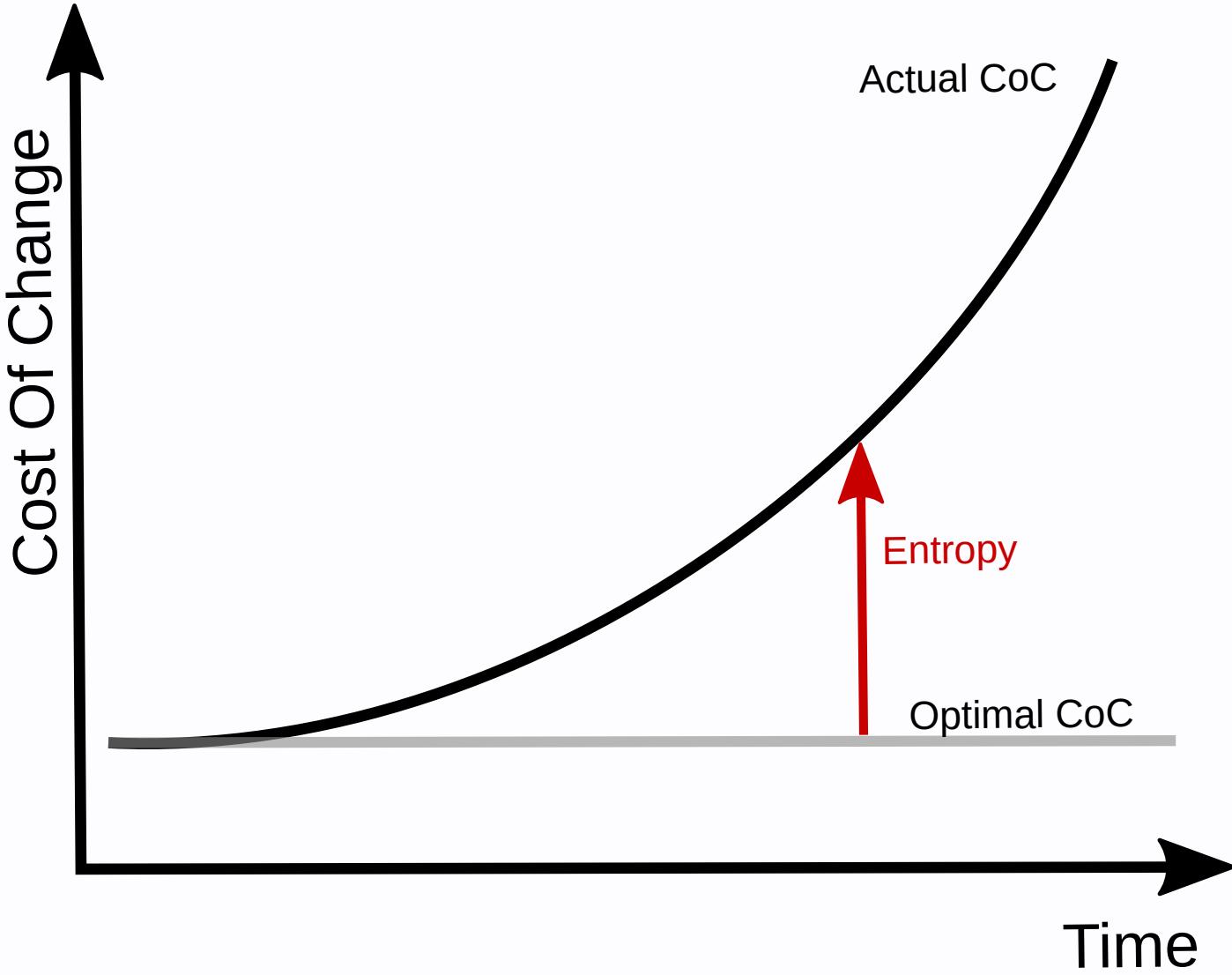
@VINCENTDNL

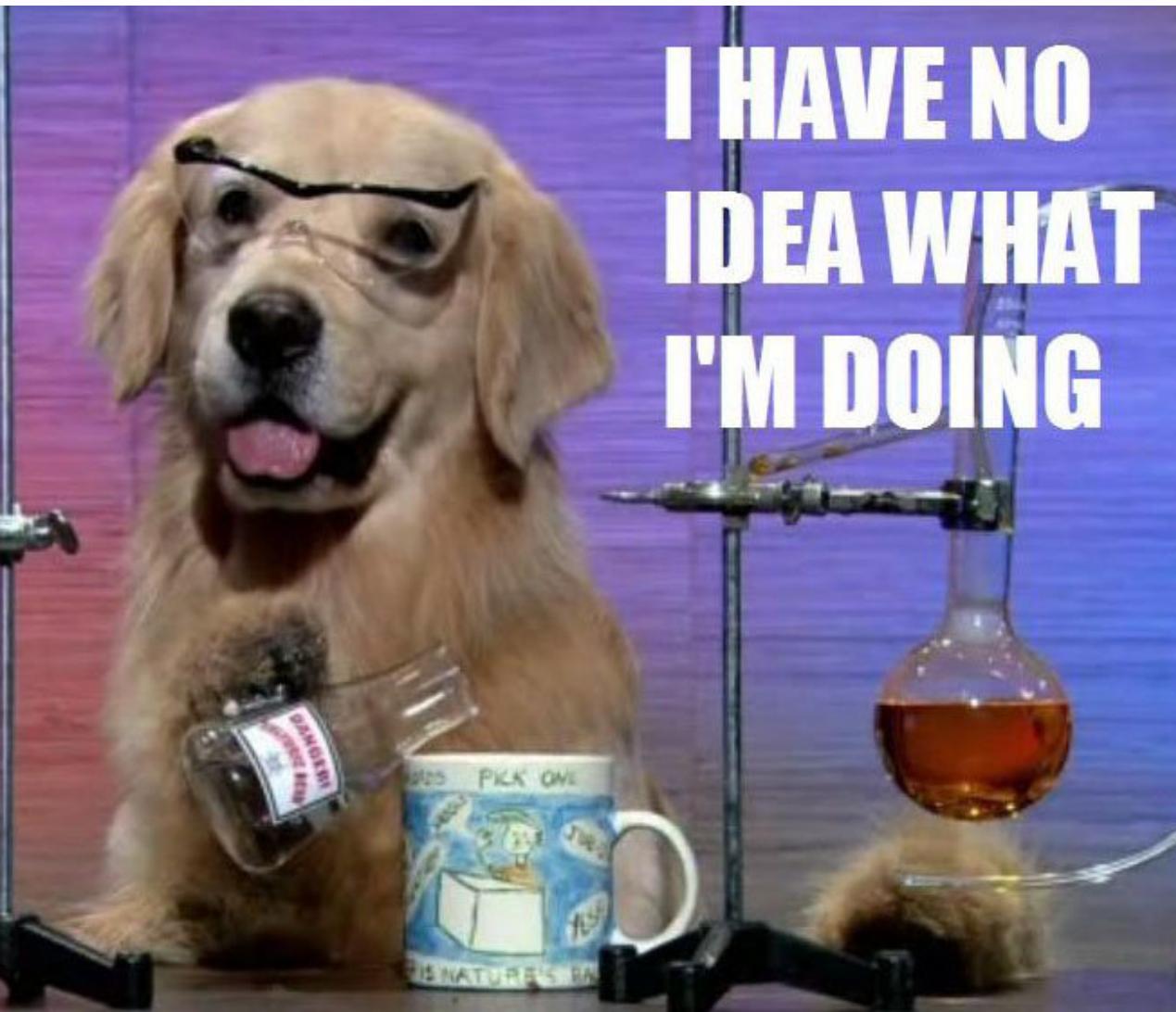


The Invisible Threat

Software Entropy

- Complexity
- Technical Debt





I HAVE NO
IDEA WHAT
I'M DOING

**What is software
development?**



Wrong understanding

- Well-defined industrial process
- Engineers and Architects
- Developers as factory workers
- More developers, more production
- Monitoring
 - number of lines
 - % comments







Proficient With Your Tools





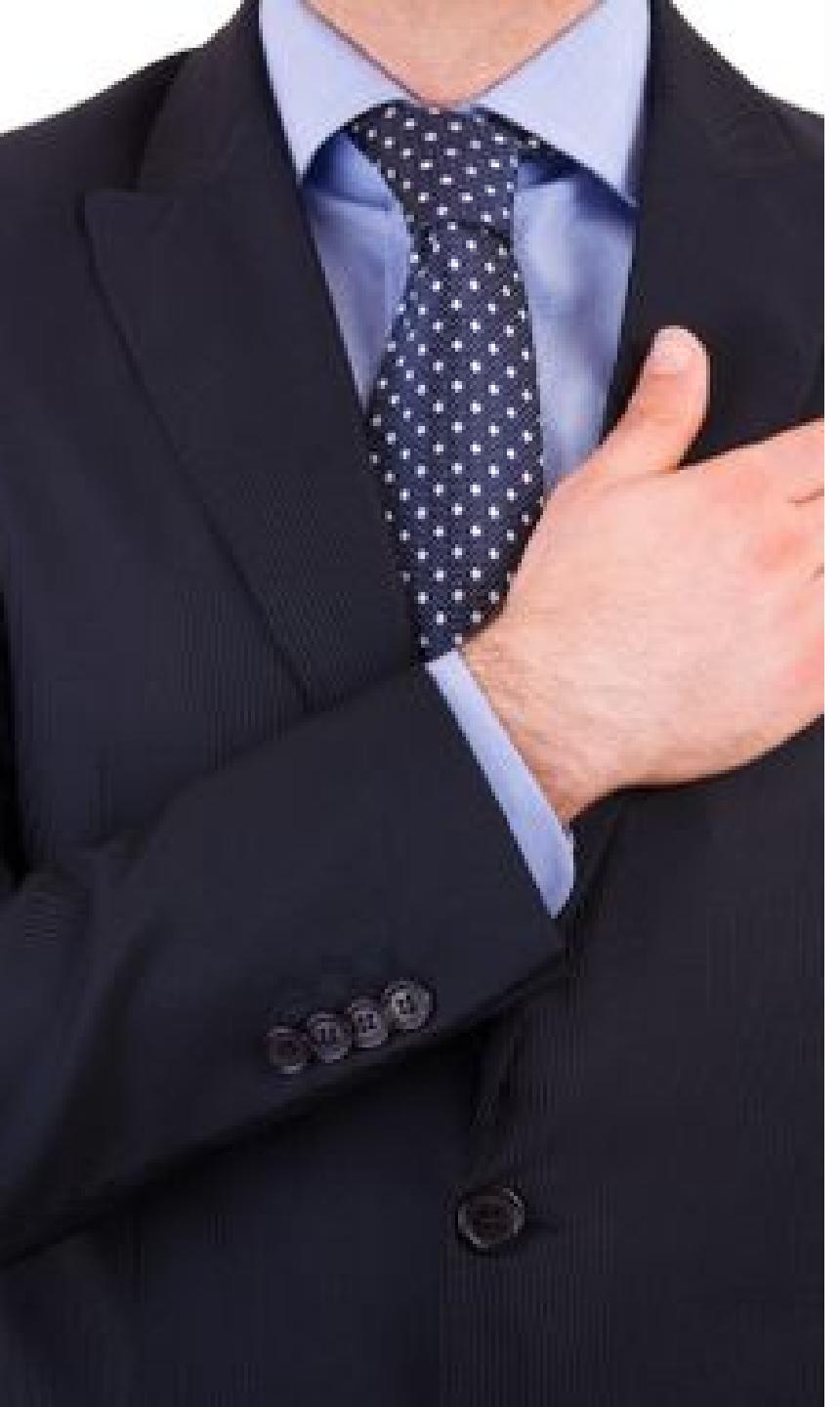
Software Craftsmanship Manifesto (2009)

Not only working software,
but also **well-crafted software**

Not only responding to change,
but also **steadily adding value**

Not only individuals and interactions,
but also **a community of professionals**

Not only customer collaboration,
but also **productive partnerships**



Mindset and Behaviors



- Be responsible & professional
- Learn how to say no
- Provide options
- Never stop learning and improving his craft
- Share your knowledges



Technical Practices

Copyrighted Material

PRENTICE
HALL

Robert C. Martin Series

Clean Code

A Handbook of Agile Software Craftsmanship

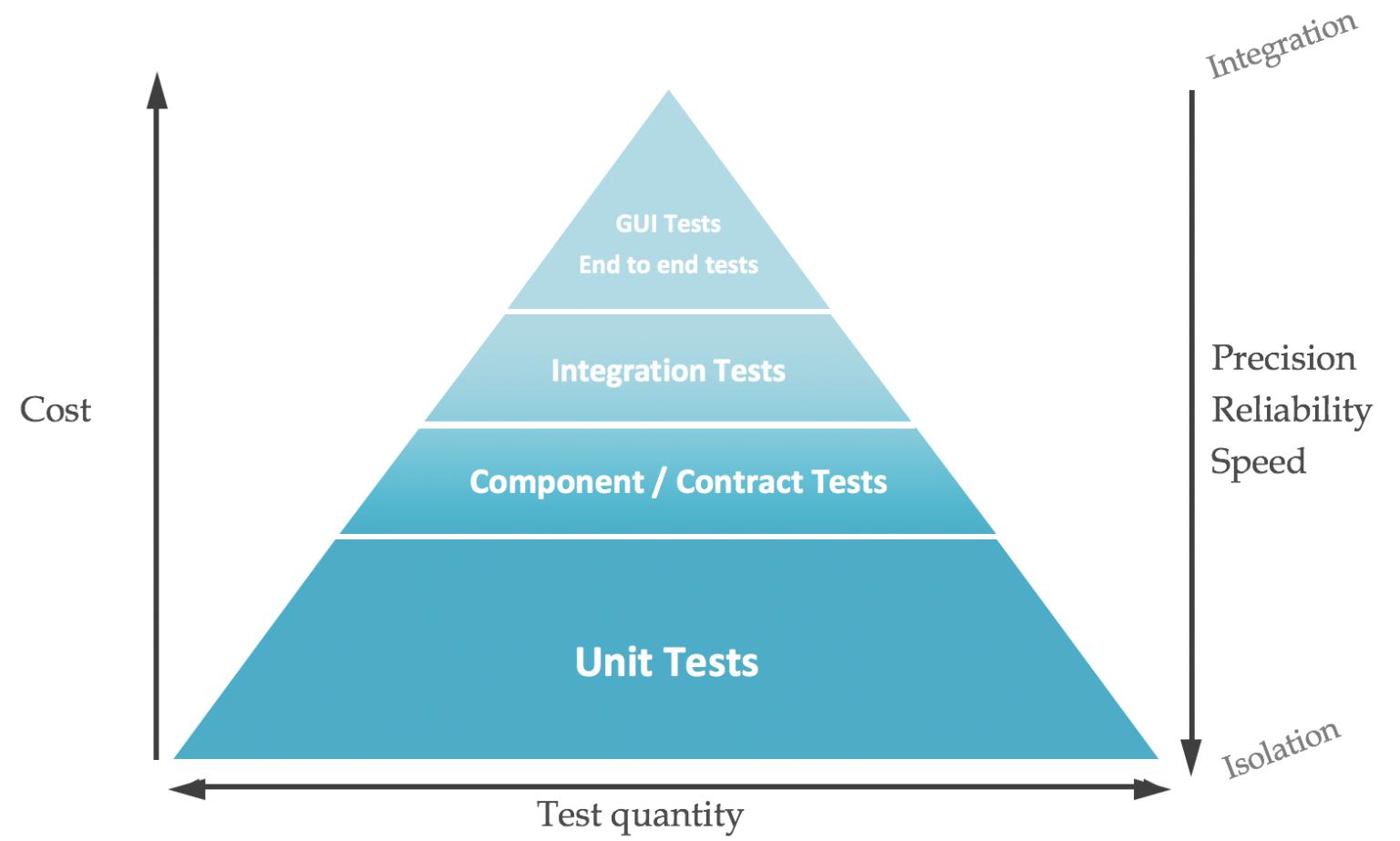
Clean code is code that is easy to understand and easy to change.

Foreword by James O. Coplien
Copyrighted Material

Robert C. Martin

Tests

- To prove a program is correct?
- To find bugs
- Construction Test
-
- l'absence de bug
- valider le logiciel?
Non, construction et trouver des bugs



Test Automation Strategy



Pair/Mob Programming

- Immediate feedback loop
- Better and cleaner code
- Collective ownership
- Share knowledge

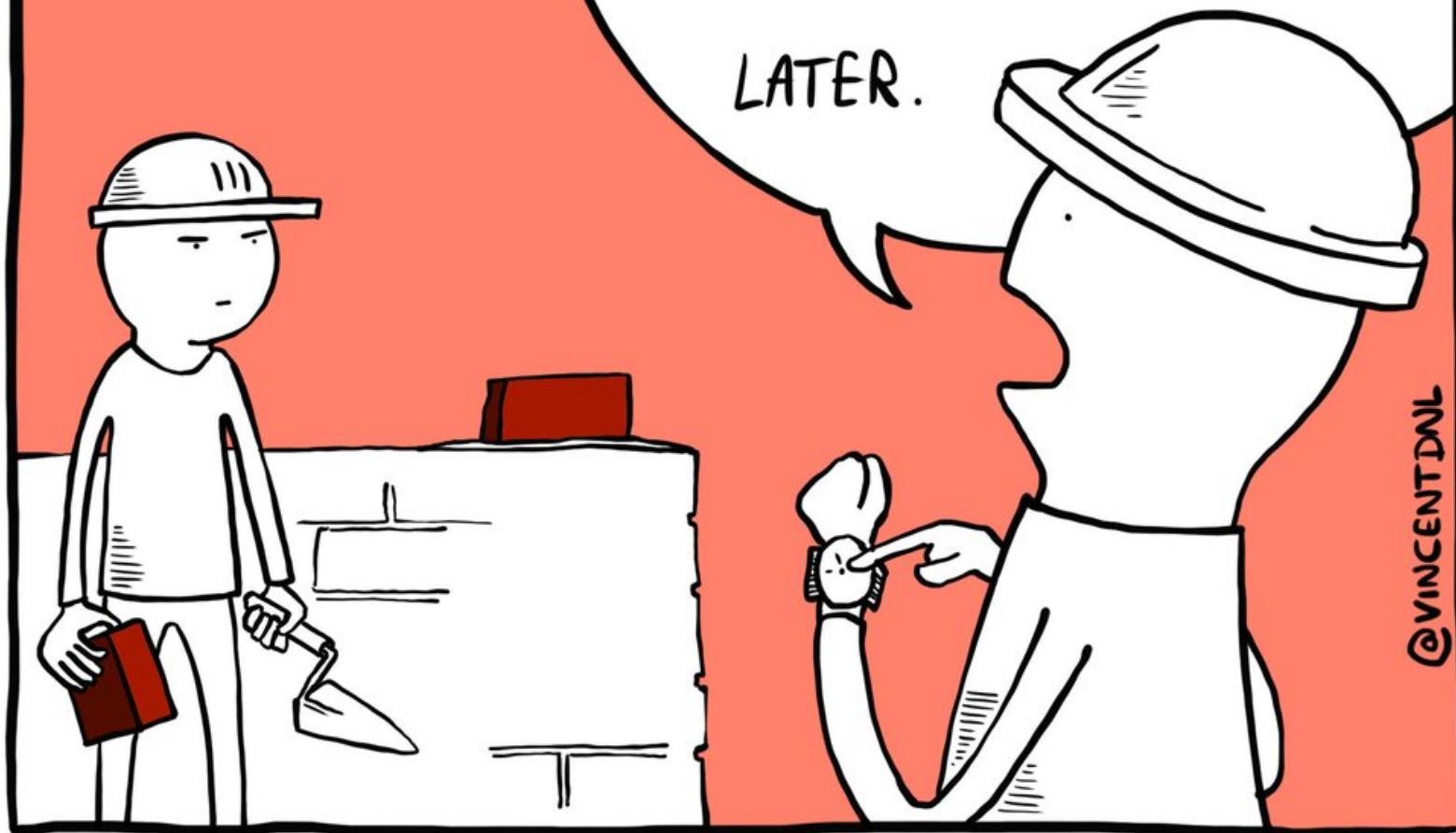
PAIR - PROGRAMMING

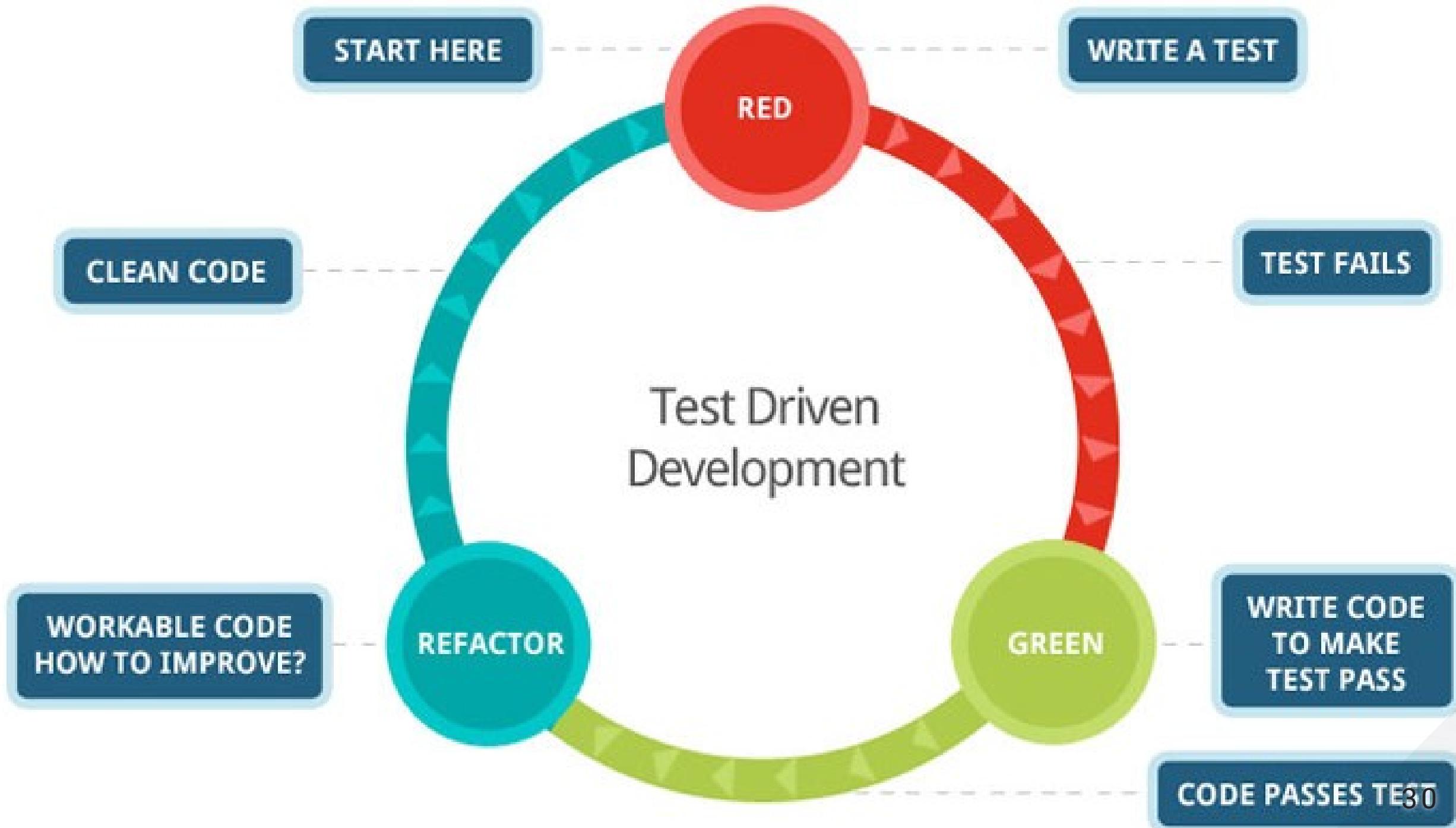
IT WOULD
GO FASTER IF
EACH ONE OF YOU
TOOK ONE.



TESTING & TDD

WE ARE BEHIND SCHEDULE
SO JUST LAY THE BRICKS
DOWN AS FAST AS POSSIBLE,
WE WILL ADD THE CEMENT
LATER.



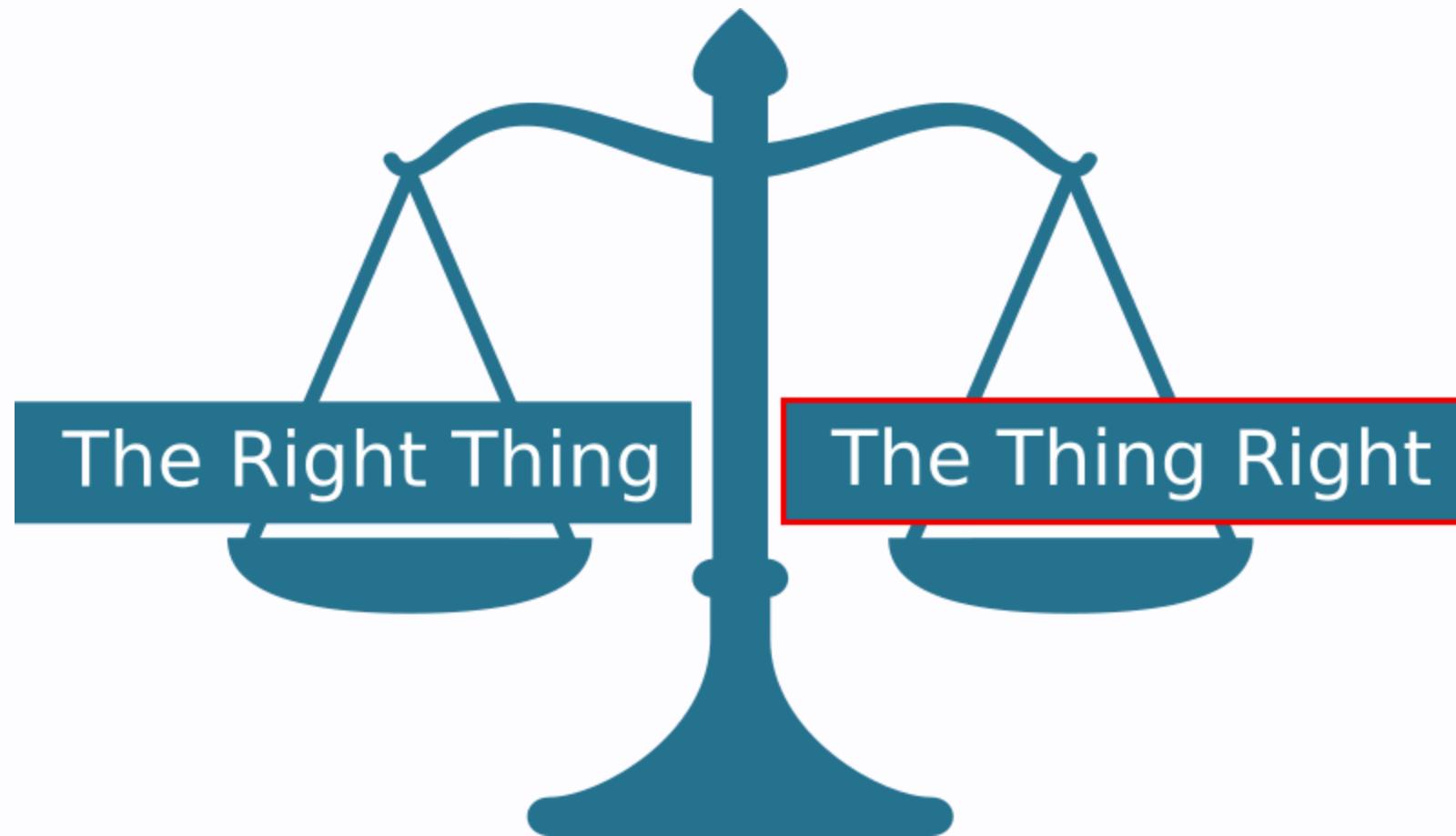




HOWTO improve our craft?

- Practice
- Practice
- Practice
- Coding Dojo, Kata...

TL;DR



“

Software Craftsmanship promises to take our industry to the next level, promoting professionalism, technical excellence, the death of the production line and factory workers attitude."

Sandro Mancuso

”

Ressources

- Book "Clean Code: A Handbook of Agile Software Craftsmanship", Robert S. Martin, 2008
- Book "Software Craftsmanship : Professionalism Pragmatism Pride", Sandro Mancuso, 2014
- Conference "The Craftsman's Oath", Robert S. Martin - SCLConf 2018