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

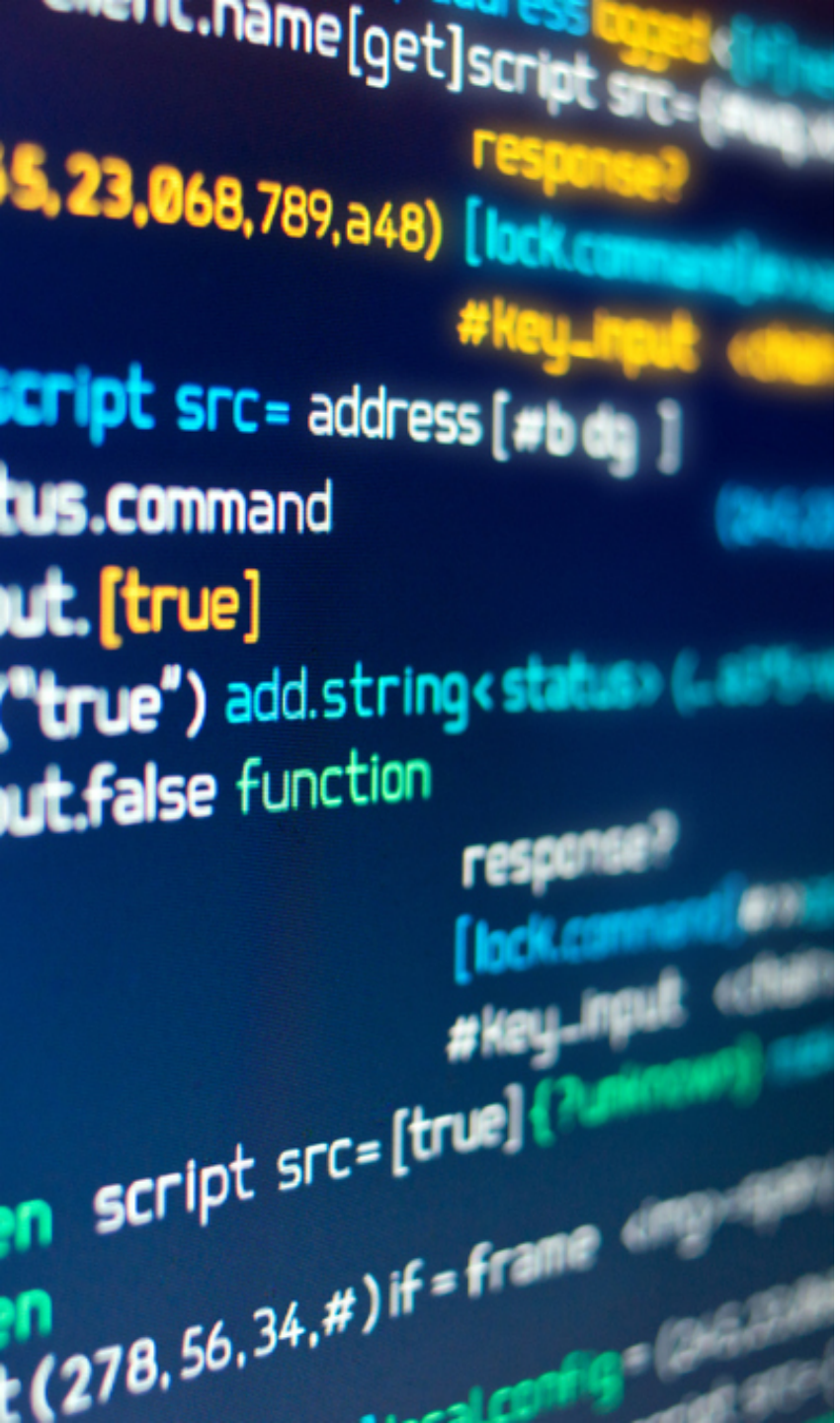
Working Software over comprehensive documentation

Customer Collaboration over contract negotiation

Responding to Change over following a plan


Stagnant Skillset
Mountain of Technical Debt
Requirements not well understood
Low Moral and Motivation
Lack of technical expertise
Long running builds
The Hangover
Unreliable Release Process
Inefficient Develop/ Debug/Deploy cycles
Unreliable and costly tests
Late discovery of bugs
Unstable system

What's Wrong?

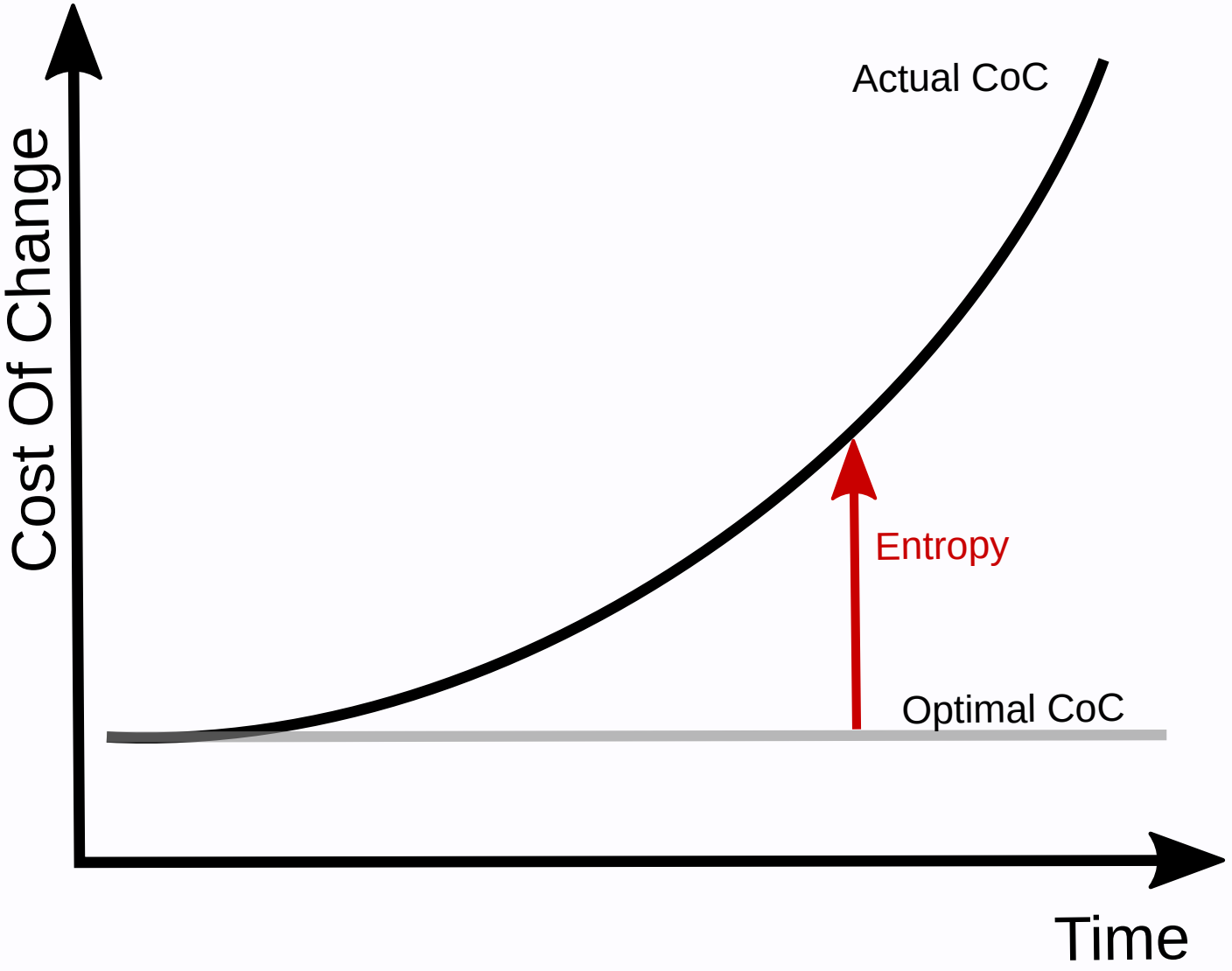


In a software product, the most important deliverable is the code itself

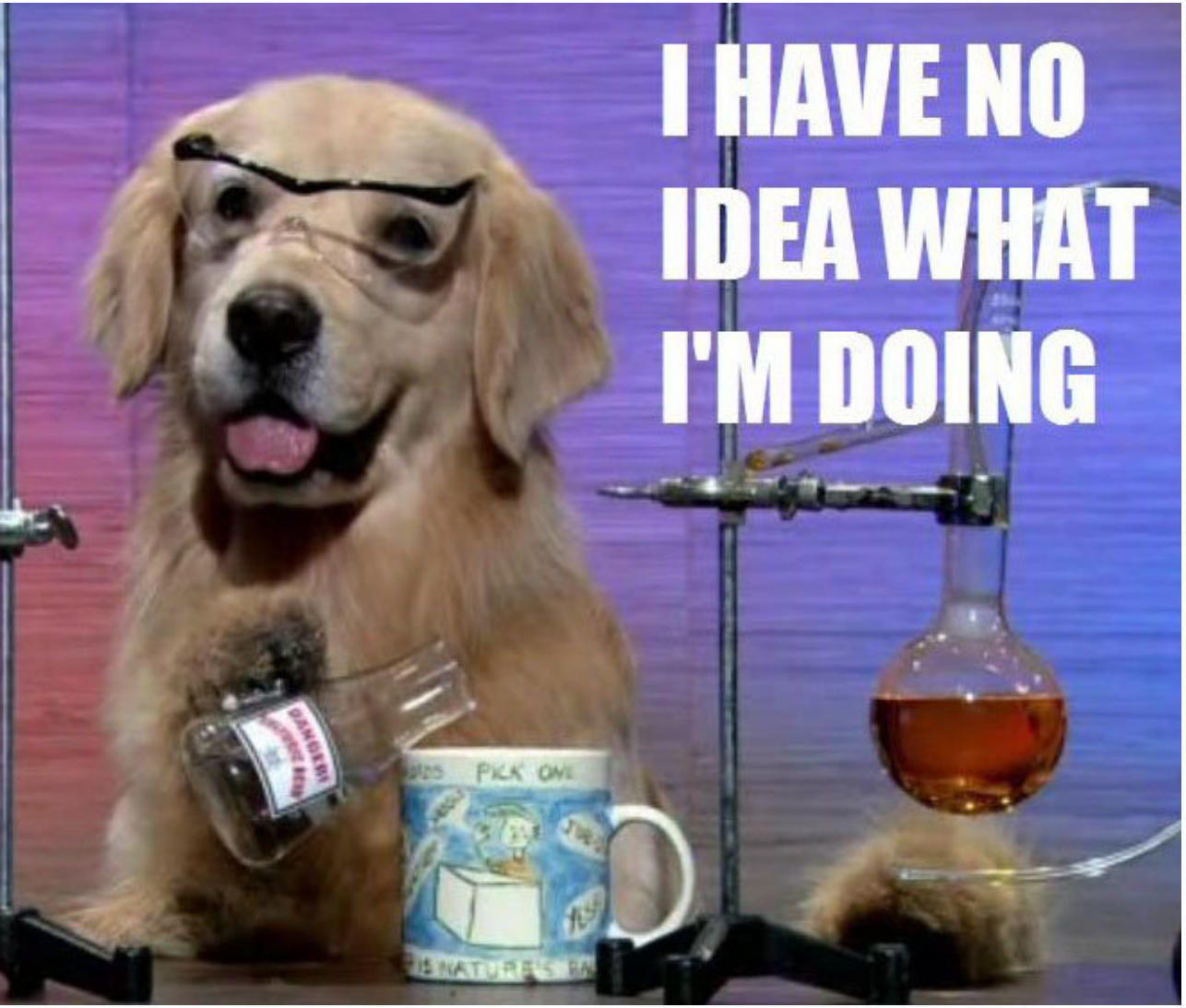
Why better code is so important?

Software Entropy

- **Complexity**
  - Essential
- **Technical Debt**
  - Involuntary
  - Voluntary


Cost Of Change
Actual CoC
Entropy
Optimal CoC
Time

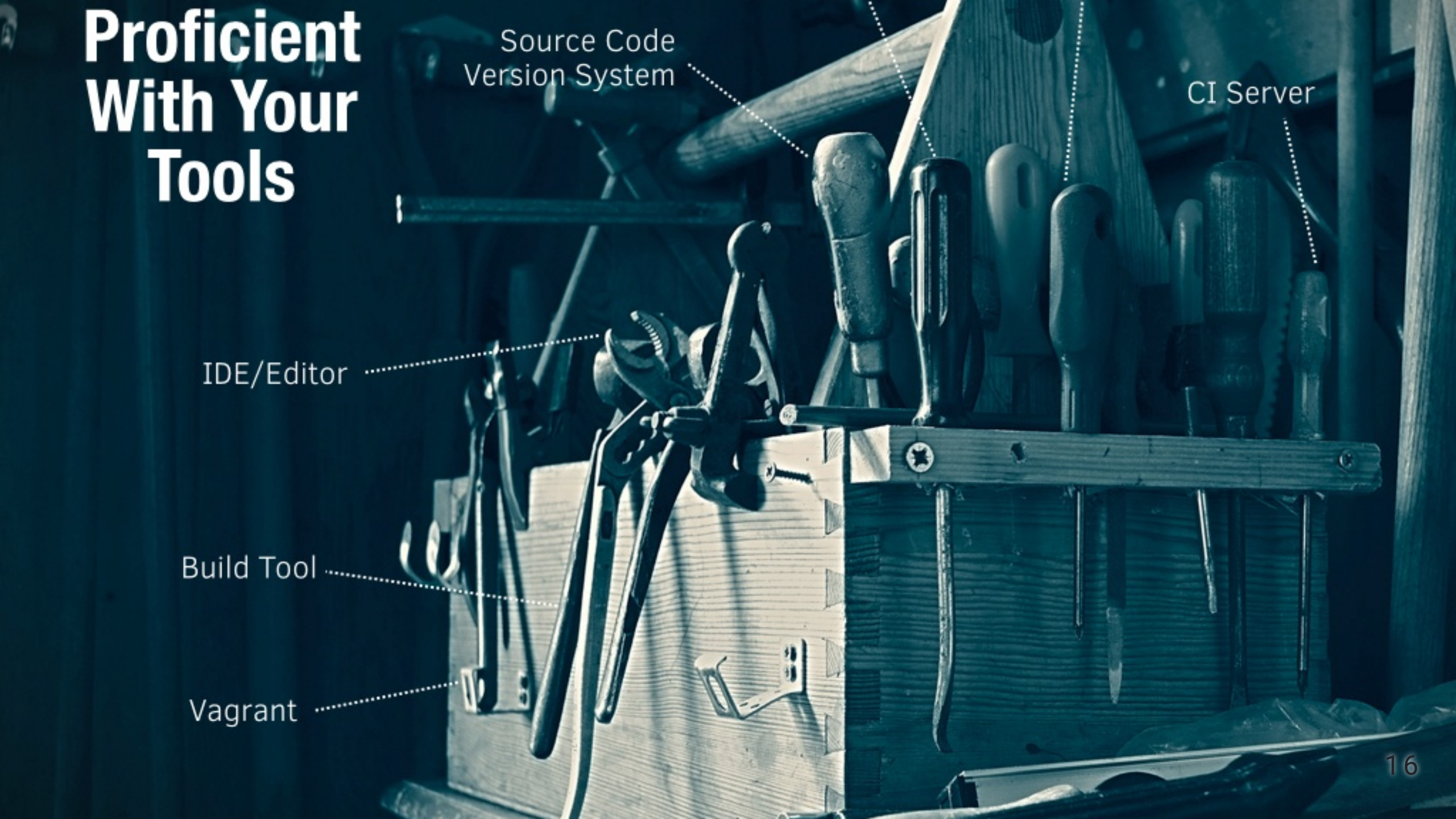

TECHNICAL DEBT
I DON'T UNDERSTAND WHY IT TAKES SO LONG TO ADD A NEW WINDOW.
@VINCENTDAIL



What is software development?

Wrong understanding

- Well-defined industrial process
- Engineeers and Architects
- Developpers as factory workers
- More developpers, more production
- Monitoring
  - number of lines
  - % comments


Proficient With Your Tools
Source Code Version System
CI Server
IDE/Editor
Build Tool
Vagrant

Software Crafsmanship 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**

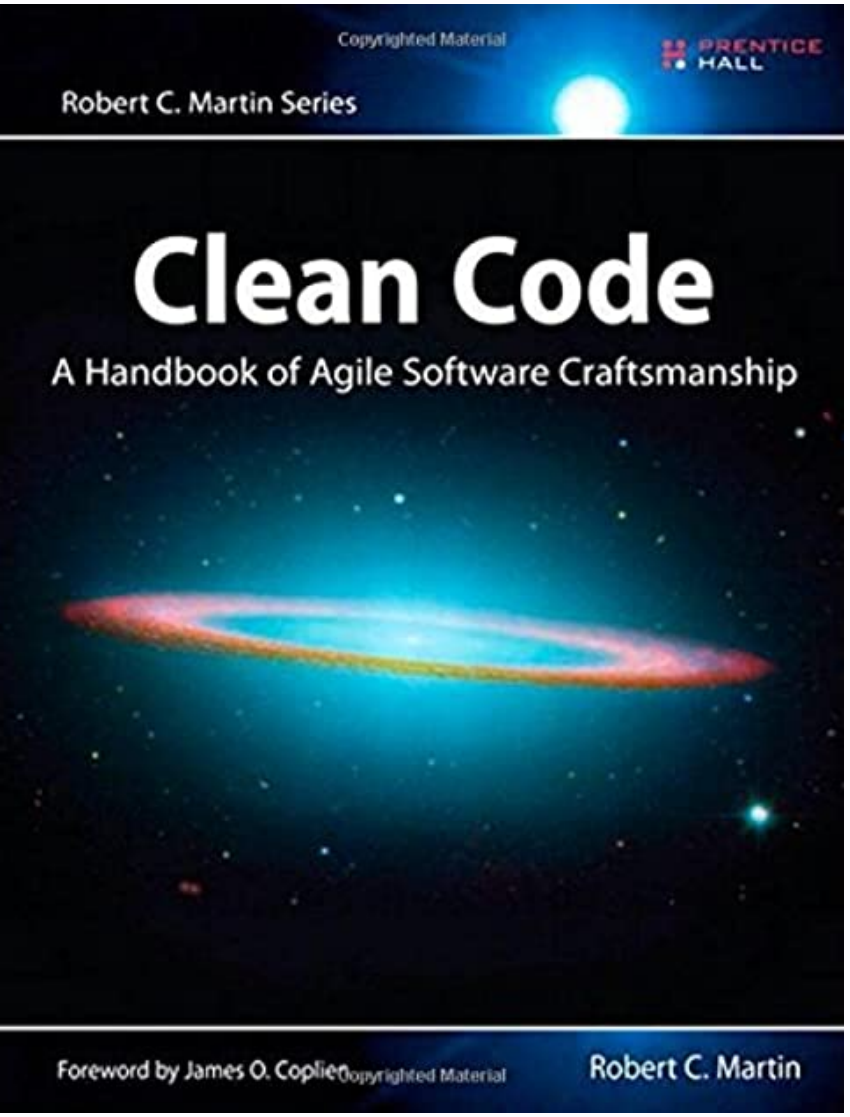
Not only customer collaboration,

but also **productive partnerships**

Mindset and Behaviors

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

Technical Practices

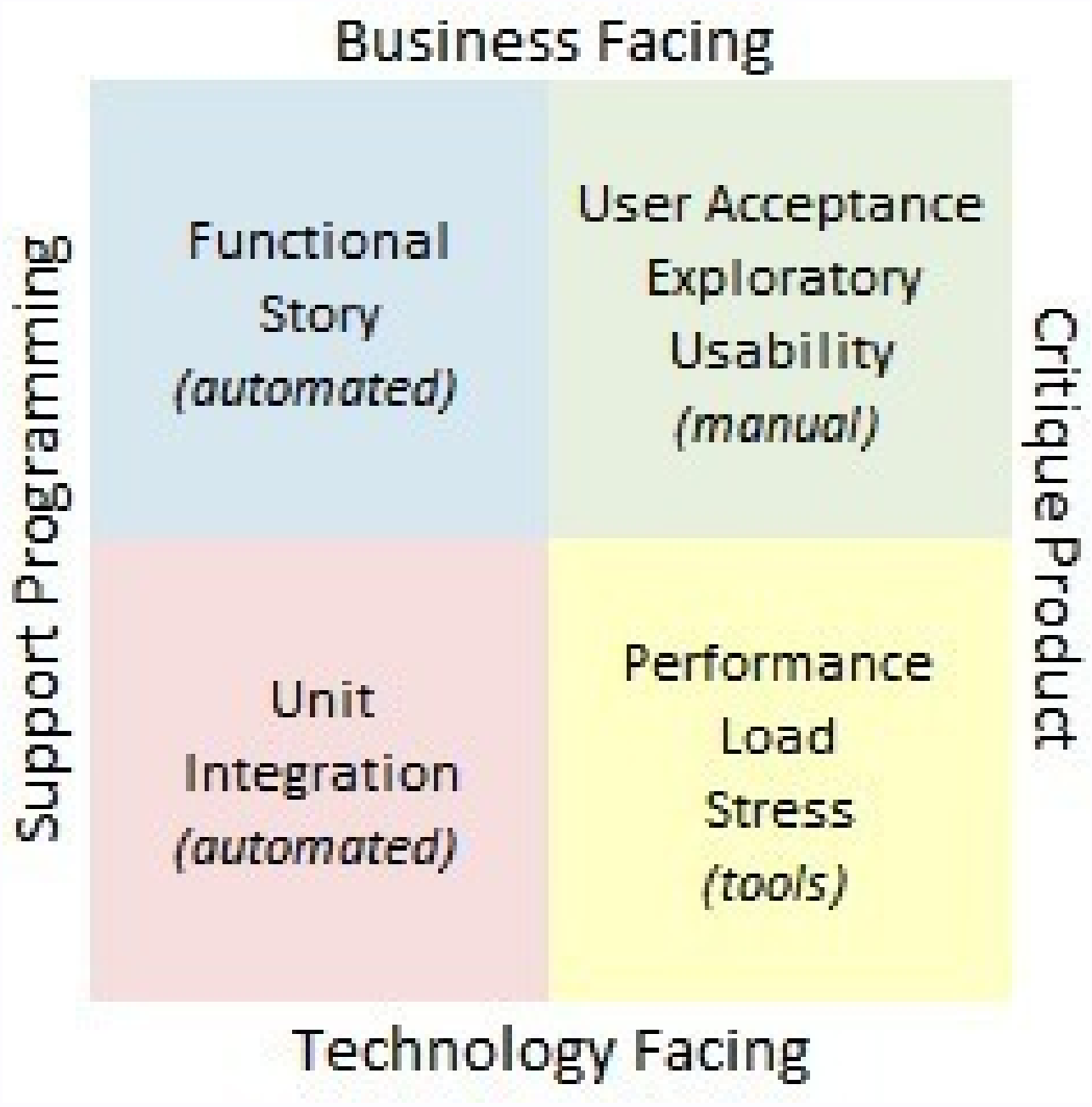


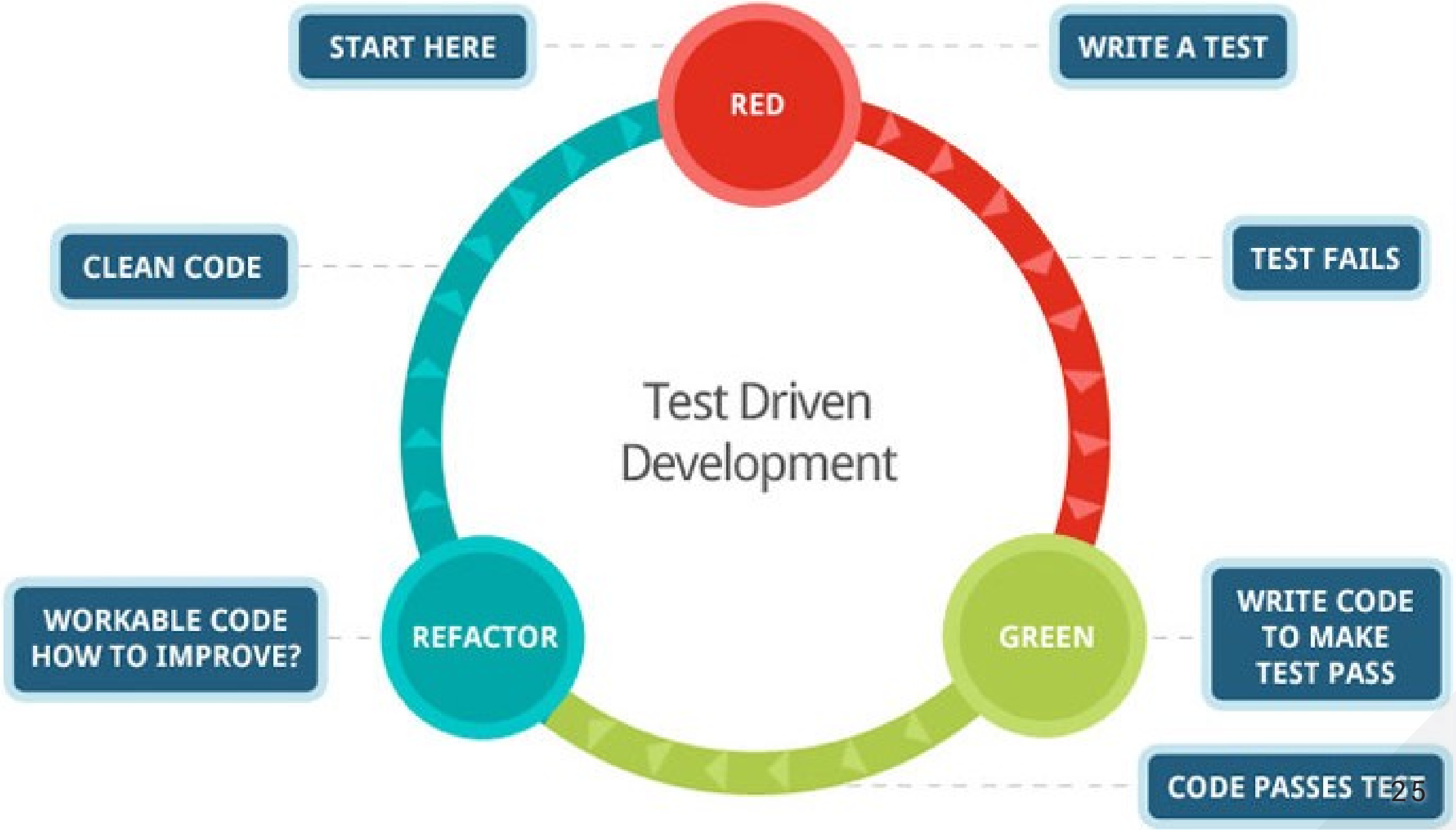
Clean code is code that is easy to understand and easy to change.

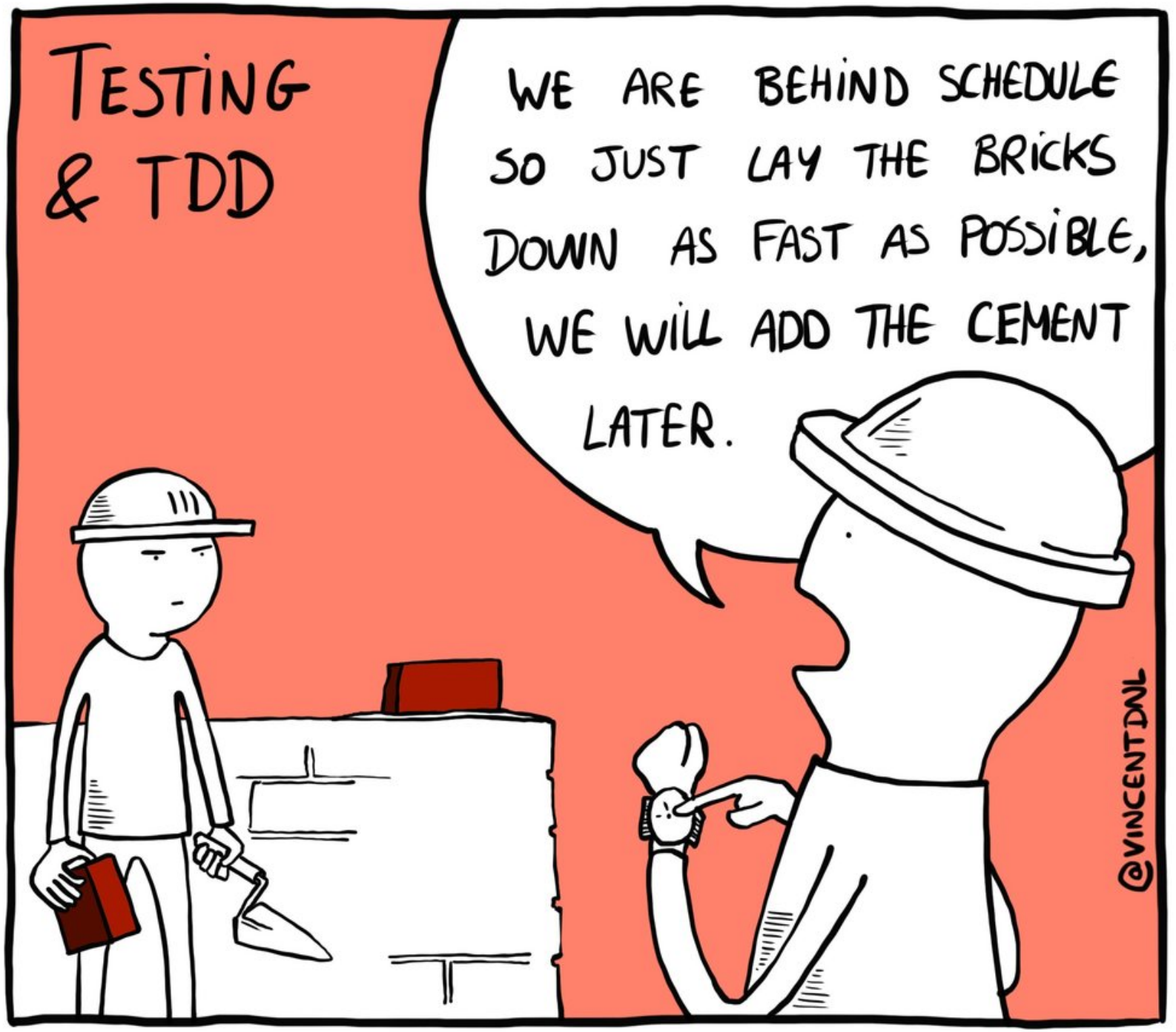
Test Strategy

Why we test?

- To prove a program is correct? NO
- To find bugs
- To build the software
- To check the software

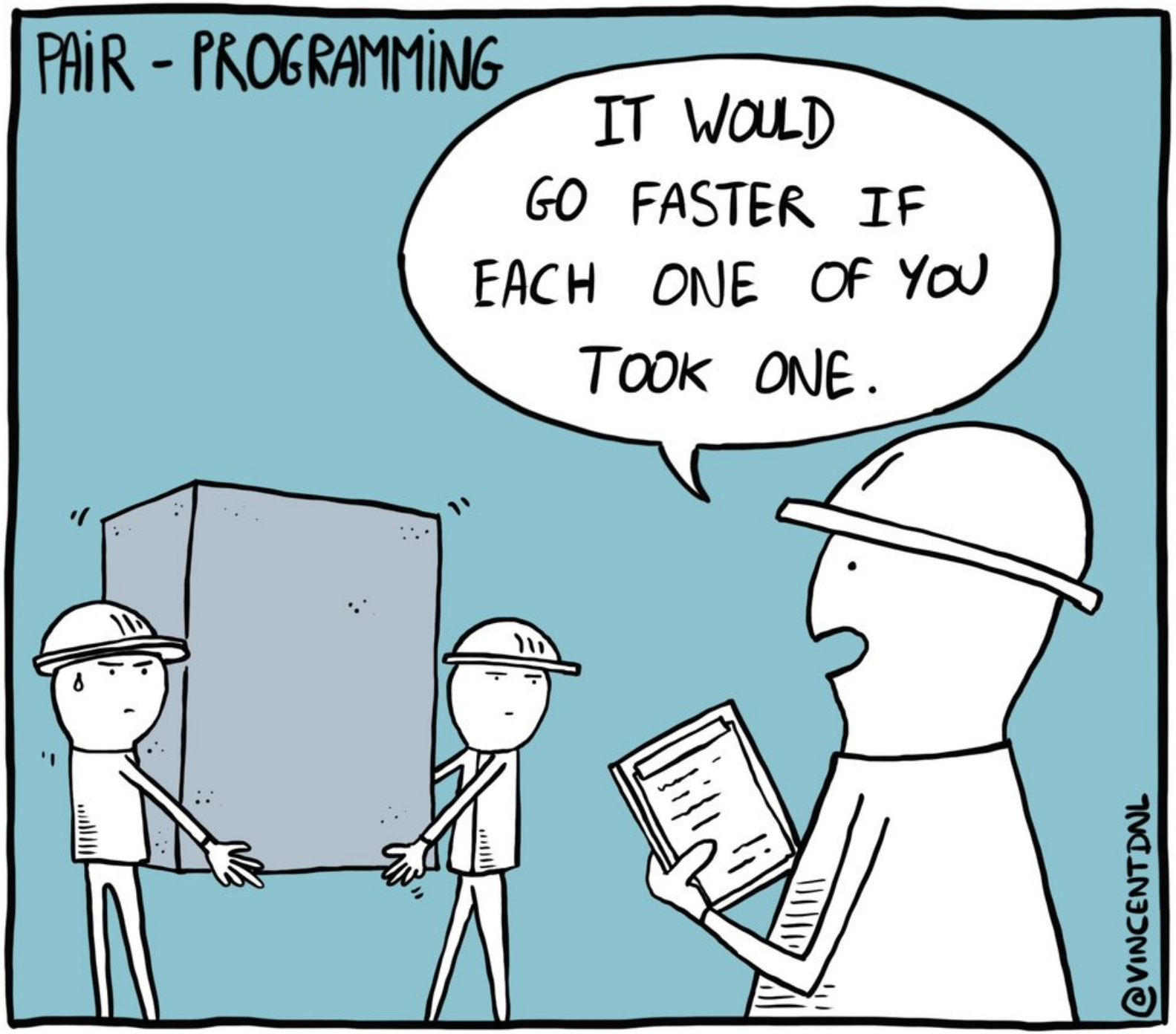

Business Facing
Functional
Story
(automated)
User Acceptance
Exploratory
Usability
(manual)
Unit
Integration
(automated)
Performance
Load
Stress
(tools)
Support Programming
Critique Product
Technology Facing


START HERE
WRITE A TEST
RED
CLEAN CODE
TEST FAILS
Test Driven
Development
WORKABLE CODE
HOW TO IMPROVE?
REFACTOR
GREEN
WRITE CODE
TO MAKE
TEST PASS
CODE PASSES TEST


TESTING & TDD
WE ARE BEHIND SCHEDULE SO JUST LAY THE BRICKS DOWN AS FAST AS POSSIBLE, WE WILL ADD THE CEMENT LATER.
@VINCENTDNL

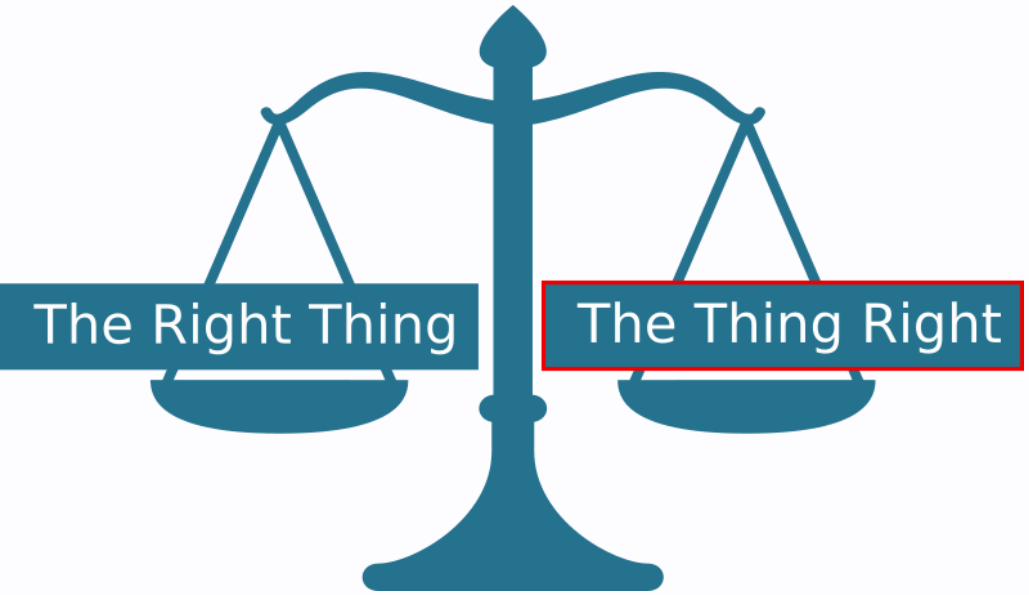
Pair/Mob Programming

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


PAIR - PROGRAMMING
IT WOULD GO FASTER IF EACH ONE OF YOU TOOK ONE.
@VINCENTDNL

HOWTO improve our craft?

- Practice Clean Code
- Practice TDD
- Practice Pair/Mob Programing
- Practice...



Conclusion

Professionalism,
Pragmatism & Pride

CCSL Software Craftsmanship Community

- DSP-FR-CCSL-CRAFT
- Salon Citadel
- Wiki