

A decade of clean code

Tips and tools for modern professional developers



—
Daniel Strmečki, JavaCRO, Rovinj, 08.05.2018
daniel.strmecki@ecx.io, daniel.strmecki@hr.ibm.com

Who are we



20
years experience

The graphic features the number '20' in a large, white, sans-serif font. To its left is a stylized outline of a shield with a star in the center. Below the '20' is the text 'years experience' in a smaller, orange, sans-serif font.

Part of
IBM iX



350+
entrepreneurs, thinkers,
geeks – people who care

The graphic features the number '350+' in a large, white, sans-serif font. To its left is a stylized outline of a tree or organizational chart with several nodes. Below the '350+' is the text 'entrepreneurs, thinkers, geeks – people who care' in a smaller, orange, sans-serif font.



7
locations

Düsseldorf, Wels,
Vienna, Bracknell,
Zagreb, Varaždin,
Bangalore.

The graphic features the number '7' in a large, white, sans-serif font. To its left is a stylized outline of a location pin. Below the '7' is the text 'locations' in a smaller, orange, sans-serif font. Below that is a list of seven cities: Düsseldorf, Wels, Vienna, Bracknell, Zagreb, Varaždin, and Bangalore, in a smaller, white, sans-serif font.



**Strategic
Platforms**

The graphic features the text 'Strategic Platforms' in a large, white, sans-serif font. To its left is a stylized outline of a rocket ship.

Adobe
SAP Hybris
Sitecore



**Any fool can write code that a
computer can understand.**

**Good programmers write code
that humans can understand.**

– Martin Fowler

What is clean code?



Clean code is simple and direct. Clean code reads like well-written prose.

– Grady Booch



**Clean code always looks like
it was written by someone
who cares.**

– Robert C. Martin



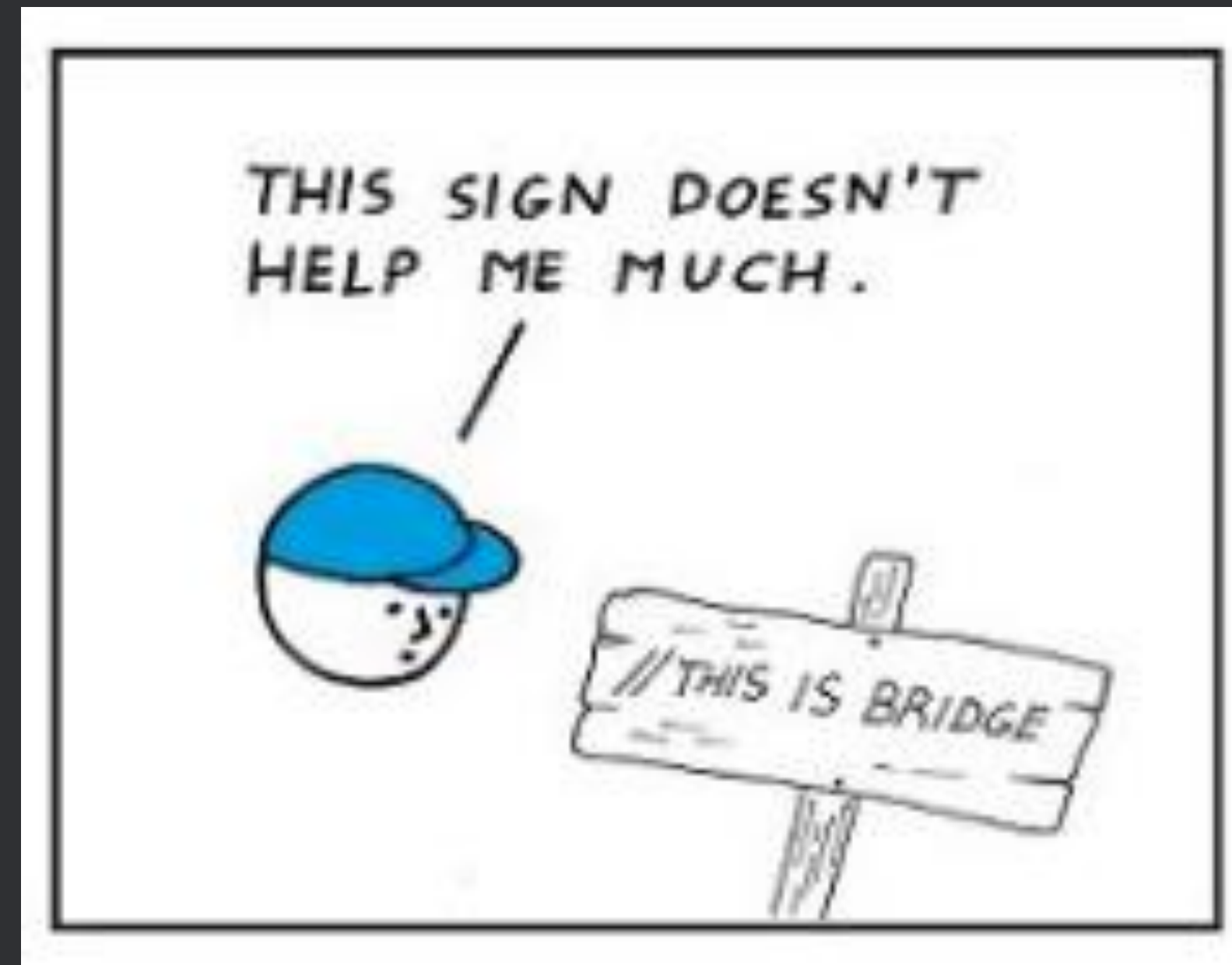
Code is clean if it can be understood easily – by everyone in the team.

With understandability comes readability, changeability, extensibility and maintainability.

What causes bad code?

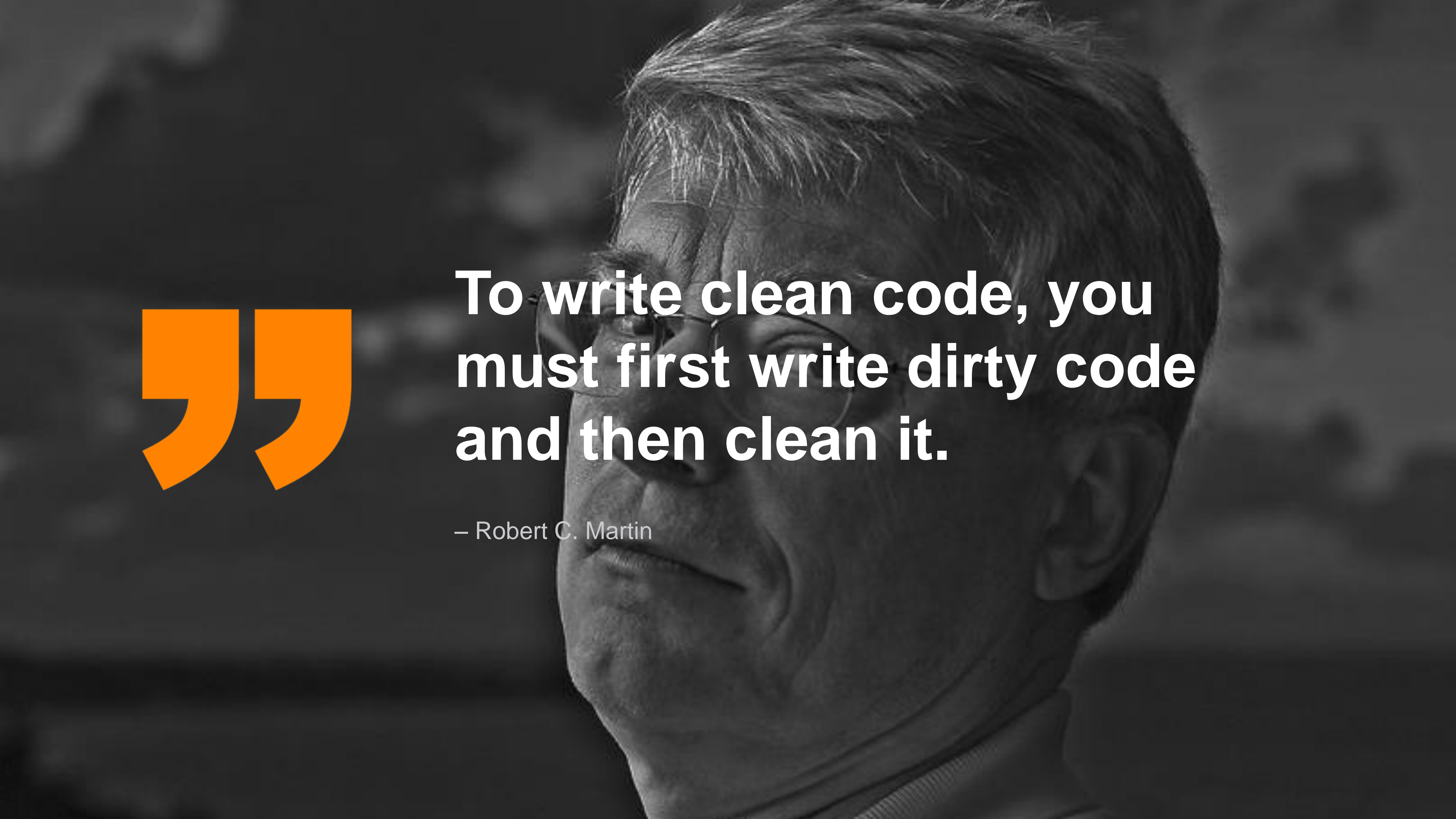
Bad code

- Bad team setup
 - Lack of experienced developers
 - Lack of specific skills
- Do it fast
 - We need this feature yesterday
 - Fast time to market
 - Competition already has it
- Bad planning
 - Low budget
 - Tight schedules
 - Angry boss





”



**To write clean code, you
must first write dirty code
and then clean it.**

– Robert C. Martin

Bad code

- Requirements change and do not meet the current solution design
- No planned time for refactoring, just new features and bugfixes
- Just get it working now, we will clean it up later
 - Everyone does it!





**LeBlanc's law: Later
equals never.**

– Robert C. Martin



Whose fault is bad code?

Management? Sales? Customer?

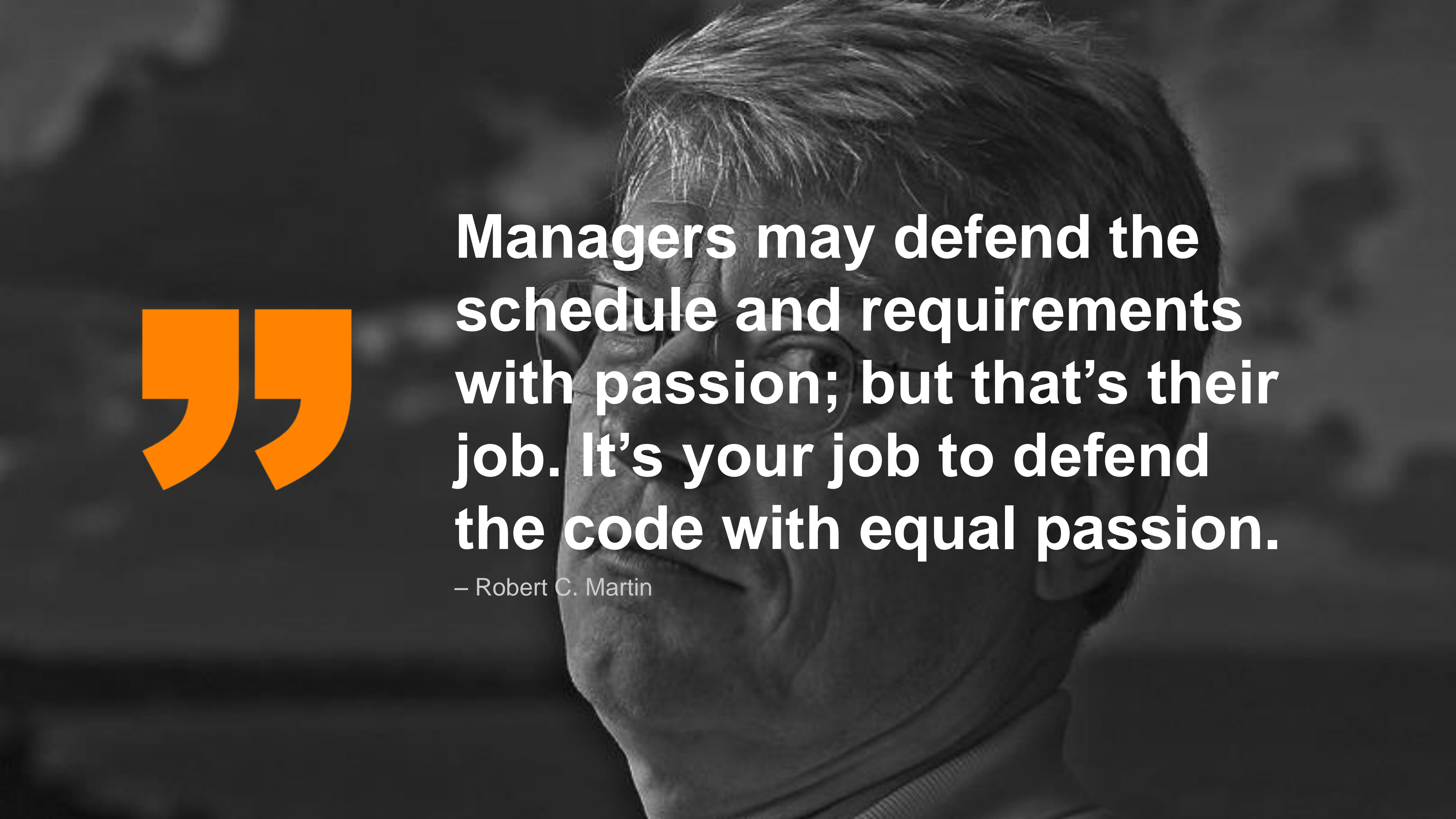
Its us, the developers

- Project is a team effort and developers are in charge of the code
- Our sales and managers ask us for information
 - If they don't, you need to make yourself noticed!
- New requirements need to fit the system design
 - Users / Requirements analysts come to us for advices
 - If they don't, you need to make yourself noticed!
- New requirements / changes / bugfixes need to fit into our schedule
 - Project managers / Scrum masters ask us for help with scheduling
 - If they don't, you need to make yourself noticed!





“



Managers may defend the schedule and requirements with passion; but that's their job. It's your job to defend the code with equal passion.

– Robert C. Martin

What is professionalism?

Professionalism

- A professional software developer
 - Is actively interested into his or her profession
 - Steadily evaluates his results and is willing to develop both himself, his team and his profession
 - Has an inner value system
 - Against this value system he double-checks his results and actions
 - Strives to stick to his values also under adverse circumstances like pressure from customers or manufacturers
 - Is not simply satisfied when his boss or his customer is satisfied



**Smart vs Professional
developer?**

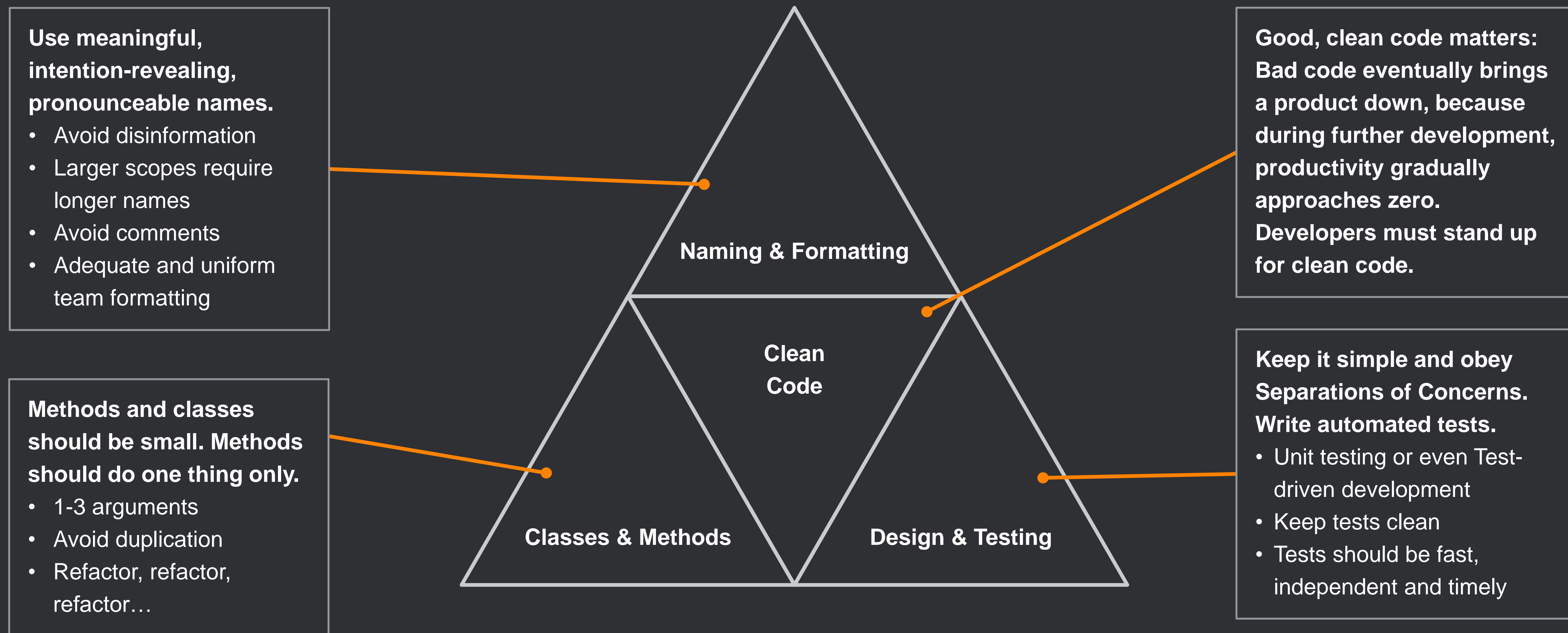
Smart vs professional developer

- A smart software developer
 - Has great developing and problem solving skills
 - Solves complex problems, writes endless lambda expressions and extremely generic code
 - Queue of developers waiting for explanation regarding how to use his code
- A professional software developer
 - Writes readable and maintainable code
 - Writes code that others can understand and reuse
 - Understands that clarity is king
 - Questions requirements and makes recommendations to the business



What have we learned?

Clean code basics



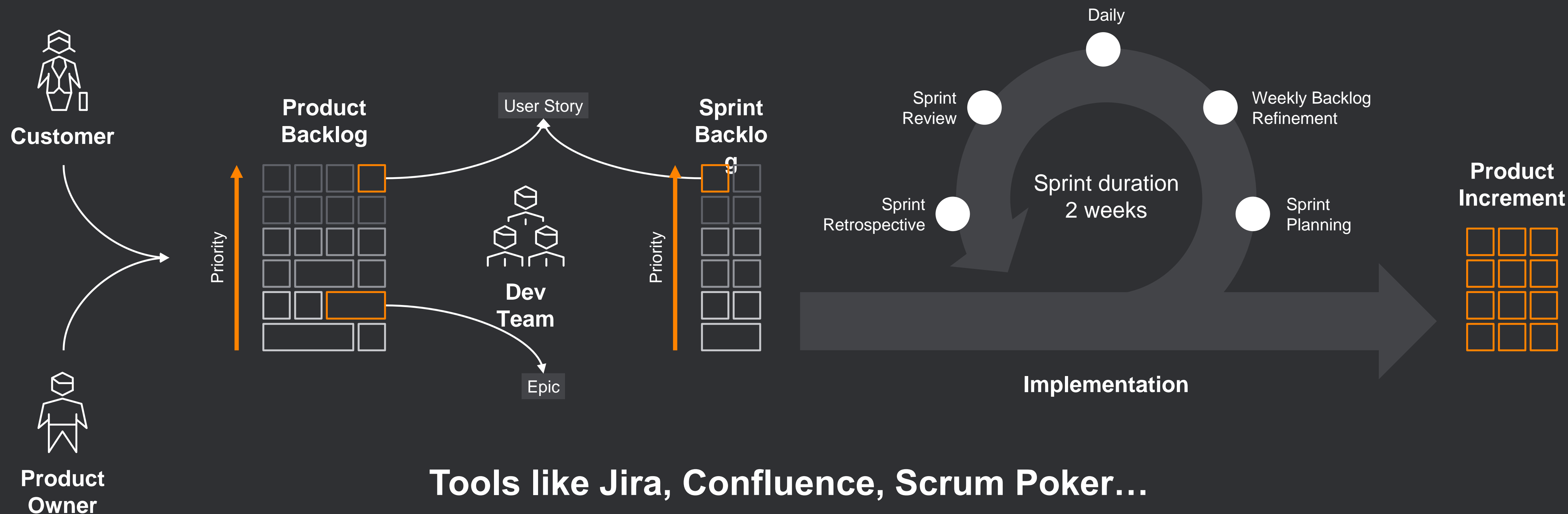


**You should name a variable
using the same care with which
you name a first-born child.**

– Robert C. Martin

**Methodologies and tools
that can help us?**

Agility is key – Scrum framework



Tools like Jira, Confluence, Scrum Poker...



Development processes and tools

Peer review process
(BitBucket or GitHub)

Style checkers
(ESLint or Checkstyle)

Static code analysers
(PMD, FindBugs or SoarQube)

Continuous integration
(TeamCity, Jenkins)

Quality KPIs
(code duplication, cyclomatic complexity)

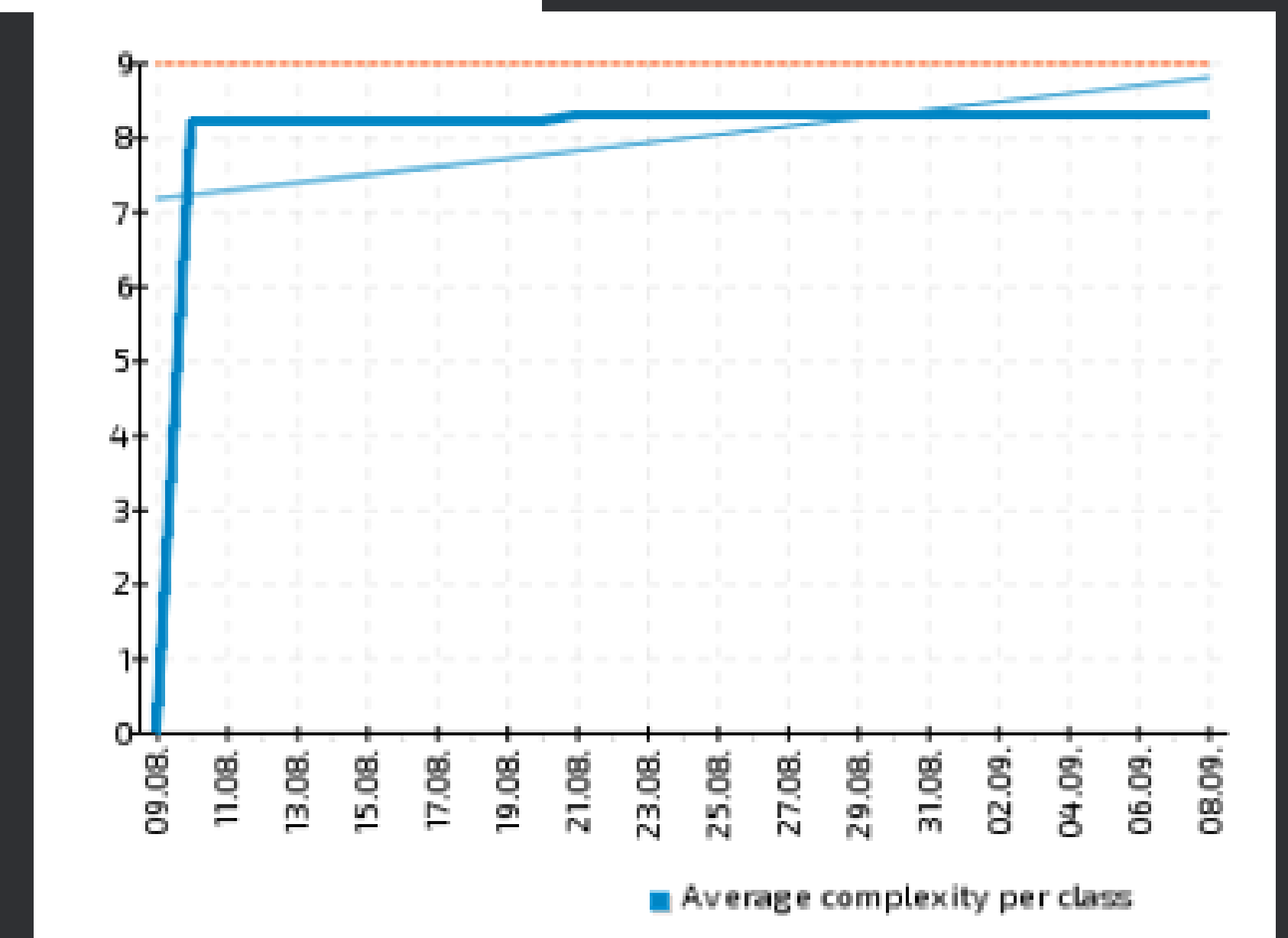
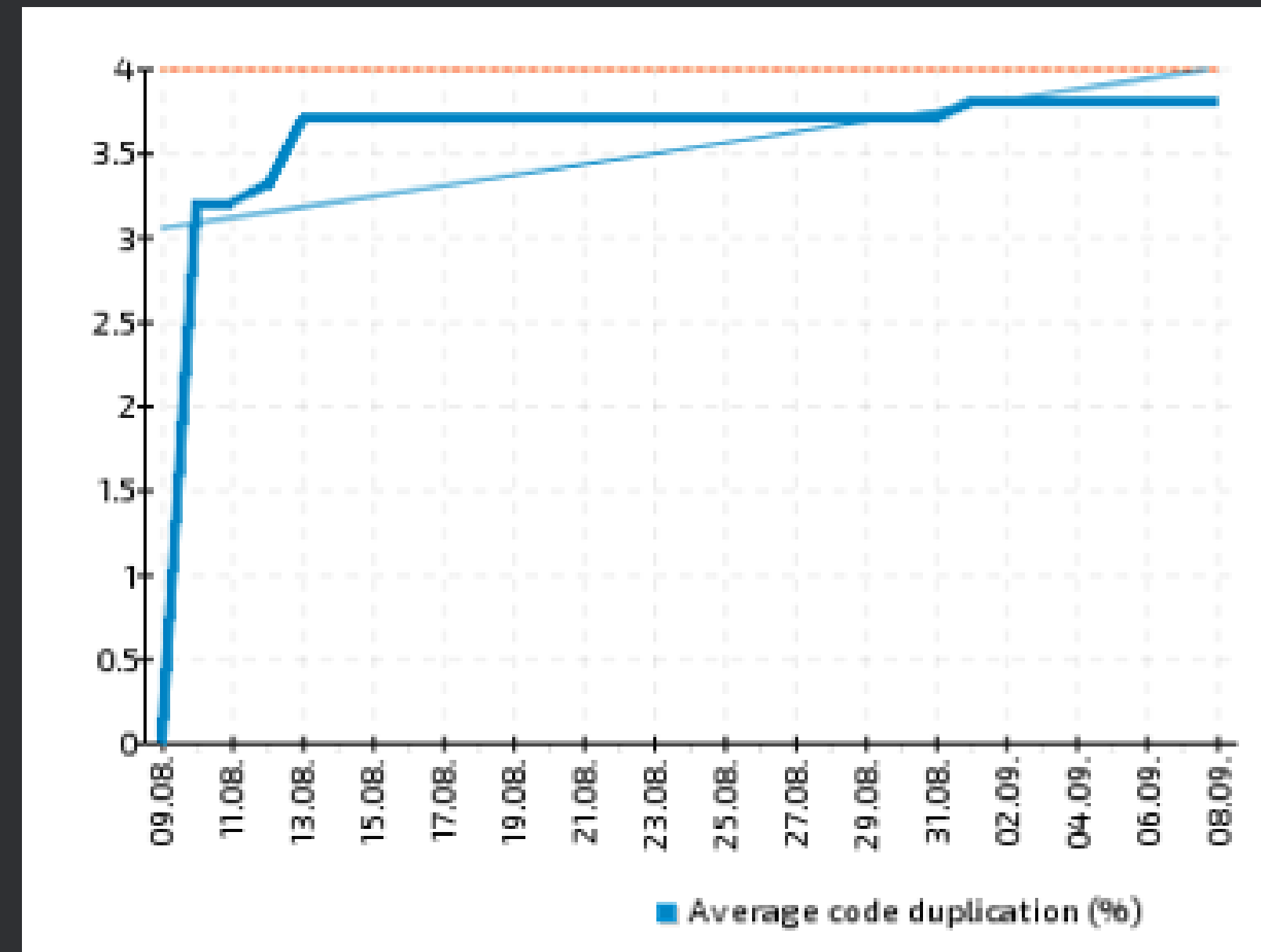


Peer review process

- Needs to be done professionally
 - Don't be offensive and don't get offended
 - Your comments should be supported by facts and examples
- Take care of
 - Code formatting and naming conventions
 - Coding best practices
 - Design patterns
 - Separation of Concerns
 - Use framework features
 - Follow the defined architecture
- And
 - Non-functional requirements
 - Maintainability
 - Readability, Configurability
 - Reusability & Reliability
 - Extensibility & Security
 - Performance & Scalability
 - OO Analysis and Design Principles
 - Single Responsibility Principle
 - Open Closed Principle
 - Dependency Injection

Quality KPIs

- We need to measure code quality
 - Number of Sonar Issues
 - Test Coverage
- And we need to set thresholds
 - Duplication < 5%
 - Complexity < 10
- Competition-driven software development boosts software quality
 - Instead of punishing Bad, reward Good projects / developers





Summary

- Clean code
 - Reads like well-written prose
 - Written by someone who cares
- Professional developers
 - Understand that project is a team effort and developers are in charge of the code
 - Understand that clarity is king
 - Make themselves noticed
 - Question requirements and make recommendations to the business
 - Stick to their values, also under pressure
- Methodologies and tools
 - Agile / Scrum
 - Developers do estimations
 - Plan for refactoring tasks
 - Peer review process
 - But do it professionally
 - Style checkers & Static code analyzers
 - Use them on every project
 - Setup team standards and stick to them
 - Continuous integration & Quality KPIs
 - Setup metrics and thresholds
 - Motivate and reward best teams

A grayscale photograph of a hand raised in a classroom. The hand is positioned on the right side of the frame, with fingers spread. The background is blurred, showing other people and papers on a wall. The lighting is soft, creating a bokeh effect in the upper left corner.

That's it from me. Thank you.

Questions?



Sources / Literature

- Robert C. Martin: Clean Code Video Series, Addison-Wesley Professional, Addison-Wesley, December 2016, ISBN: 0134661745
- Robert C. Martin: Clean Code - A Handbook of Agile Software Craftsmanship, Prentice Hall, Addison-Wesley Professional, August 2008, ISBN: 9780136083238
- Clean Code Developer: An initiative for more professionalism in software development, available 29 April 2018 on <http://clean-code-developer.com/>
- Jan de Vries: Clean Code, available 29 April 2018 on <https://www.slideshare.net/JandV/clean-code-summary>
- Navdeep Singh: Good code ss Bad code, available 29 April 2018 on https://medium.com/@navdeepsingh_2336/good-code-vs-bad-code-35624b4e91bc
- Wojtek Lukaszuk : Summary of Clean code, available 30 April 2018 on <https://gist.github.com/wojteklu/73c6914cc446146b8b533c0988cf8d29>
- Evoke Technologies: Code Review Checklist, available 30 April 2018 on <https://www.evoketechnologies.com/blog/code-review-checklist-perform-effective-code-reviews/>
- DZone: Java Code Review Checklist, available 30 April 2018 on <https://dzone.com/articles/java-code-review-checklist>
- Marc Kurz, David Stöger: Competition-Driven Quality Boost, Software Quality Days 2017